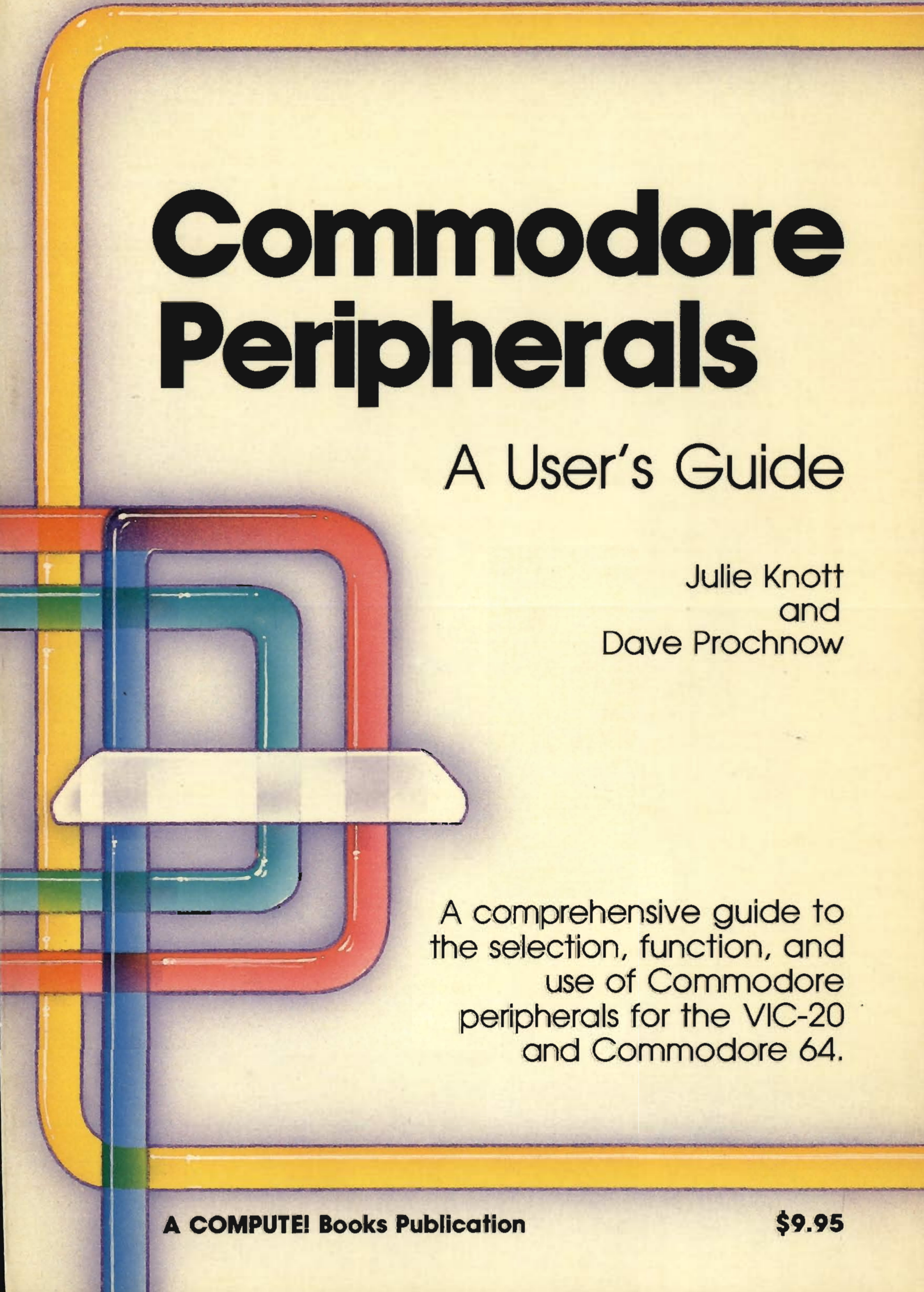
Commodore Peripherals

A User's Guide

Julie Knott
and
Dave Prochnow



A comprehensive guide to the selection, function, and use of Commodore peripherals for the VIC-20 and Commodore 64.

A COMPUTE! Books Publication

$9.95

Commodore Peripherals

A User's Guide

Julie Knott and Dave Prochnow

COMPUTE!™ Publications,Inc. abc
One of the ABC Publishing Companies

Printed in the United States of America

ISBN 0-942386-56-6

10 9 8 7 6 5 4 3 2

COMPUTE! Publications, Inc., Post Office Box 5406, Greensboro, NC 27403, (919) 275-9809, is one of the ABC Publishing Companies and is not associated with any manufacturer of personal computers. Commodore 64 and VIC-20 are trademarks of Commodore Electronics Limited.

Contents

Foreword

Wading through computer operating manuals can be frustrating. Information can be hard to find, if it's there at all. And often it's written in language too technical to understand easily.

This problem isn't limited to computers. The peripherals you need for printing out documents, communicating over the telephone, or even saving and loading software can be just as difficult to figure out.

One approach is trial-and-error. You can experiment with different switch configurations, various cable hookups, and even random commands. But that can cause problems. For example, if you use the wrong commands with a disk drive, you might hang up your system, destroy a file, or even erase a disk completely.

Commodore Peripherals: A User's Guide is the alternative. It explains the operation and use of many readily available Commodore peripherals in easy-to-understand terms. Each chapter is devoted to one type of peripheral—printers or modems, for example—and the major functions and commands are clearly documented. You'll also find program examples that show you how to use peripherals in your own programs.

Chapter 1 gives you an overview of the VIC and 64 and explains the input/output ports and their functions. Chapters 2–10 look at specific peripherals, including some (like the CP/M cartridge for the Commodore 64 or the 1600 VICMODEM) that are not often discussed or documented.

You'll find a useful reference section at the end of each chapter, which includes a listing of the major functions that apply to the device discussed there. Most chapters include a glossary of terms for quick reference, as well as a listing of error messages and their meanings.

This book assumes no prior experience with Commodore peripherals, but it doesn't skimp on technical information. As a result, it will be useful as a quick reference book for both the beginner and the experienced user who wants to get the most from any Commodore 64 or VIC-20 computer system.

Chapter 1

The Computers

Chapter 1

The Computers

Without connections to the outside world a computer is worthless. At the very least, it must be hooked up to a power source and to a video display. It's possible to get along with only those two connections, but by connecting other devices to your computer's other ports you can make computing easier and a lot more fun.

If you think of your Commodore computer as a brain, the input/output (I/O) ports can be considered its senses. The devices you connect to those ports let your computer interact with the outside world by speaking through a TV, monitor, printer, or even a synthesized voice; by listening through a modem ear; by memorizing complex ideas on a floppy disk or cassette tape; or even by performing higher levels of thinking with memory expansion or language-enhancement cartridges.

As a programmer, you are the one who ultimately coordinates the actions of the various parts of your Commodore computer's body. That body has great potential, but until you fully activate its senses (by learning to control its many I/O ports and associated peripherals), it will never realize its full potential.

Despite obvious similarities in outward appearance, the VIC-20 and Commodore 64 are quite different internally. Therefore, it's a good idea to take an individual look at each computer's ports.

The VIC-20

Hooking Up Your System

To prepare your VIC system for use, you'll need a VIC-20 computer, a television set, the computer's power-supply box (with attached cable), the RF modulator, the video cable, the switch box, and a small screwdriver.

On the back of your TV, locate the two screws that are labeled VHF. Ignore the screws labeled UHF. Ordinarily, those screws let you connect a TV antenna for better reception of TV broadcasting signals. But by removing the antenna and replacing it with the wires from the computer/TV switch box, your

TV will be able to accept signals from your computer instead.

Here's how to do it. Loosen the two screws marked VHF and remove any existing antenna leads. Then slip one lead from the computer's switch box (the flat metal box with a sliding switch and positions marked COMPUTER and TV) underneath each VHF screw. It makes no difference which lead goes to which screw. Then, using a small screwdriver, tighten the VHF screws over each lead.

You may use either a color or black-and-white television with your VIC-20. The video signal will be accepted in either case, but some pieces of software that utilize certain color combinations might become illegible on a black-and-white television screen.

Now, you're ready to connect the video cable, a short black cord with a phono plug at each end. Plug one end into the switch box and the other end into the RF modulator, a small metal box that contains the circuitry which converts the signal from your VIC-20 into a signal that can be decoded by your TV to produce a picture on the screen.

The cable permanently attached to the RF modulator box must be plugged into the VIC-20's video port. Viewing the VIC from the rear, the video port is the leftmost of two round sockets on the rear panel.

You need to make one simple adjustment in order for this video arrangement to work: Select the TV channel you want to use and then set the RF modulator channel select switch accordingly. Choose either channel 3 or 4, whichever gives the clearest picture.

Now, connect the power supply. First, be sure that the on/off switch of the VIC-20's right side panel is off. Then, plug the power-supply cord into the power socket located to the right of the power switch. Finally, plug the power supply into any household electrical outlet. Now your system is ready to be turned on.

Always plug the computer and TV or monitor power cords into separate household electrical outlets. This helps eliminate possible interference patterns that may distort the screen image. Note, too, that all peripherals which use their own power supplies should be turned on before turning on the computer; this avoids the possibility of a power surge which could damage the circuitry of the VIC. The parts of your system should be turned on in the following order: television or monitor first, then disk drive, printer, any other peripherals, and finally the VIC-20 itself. It is equally important to turn your equipment off in the *reverse* order of that given above.

Remember to unplug the VIC-20's power supply from the household outlet when the computer is not in use. Even though the computer is turned off, the power-supply unit continues to operate and can become warm if plugged in for long periods of time. One solution to this problem is to purchase an outlet strip with an on/off switch. Plug the computer into this power strip, and after you turn off the computer, just turn off the outlet strip as well. Some outlet strips even protect your computer system from lightning and other surges in household current.

Joystick Port

The VIC-20's input and output areas can be divided into two distinct groups: those on the computer's right side panel and those on the rear panel.

The VIC-20's right side panel holds the power-cord socket, the on/off switch, and a game port (labeled control port) that accepts a joystick or set of game paddles. Commodore manufactures both devices (they are identified as the VIC-1311 Joystick and the VIC-1312 Game Paddles), but you can also use joysticks or paddles produced by other manufacturers.

You can think of a joystick as five switches. Four of them read directions and one is used as the fire button. The four direction switches, when activated individually by the joystick, yield readings corresponding to four directions—up, down, left, and right. When the stick is pushed diagonally, two directions are read at once. The VIC-20 itself does not have the ability to take true diagonal joystick readings, but it can be done by using additional programming.

Joysticks are read in BASIC through a series of PEEKs. Because you'll want to read each switch individually, a series of IF-THEN statements looks for one of the five unique numbers that represent each direction and the fire button.

```
10 X=PEEK (37137)AND126
20 POKE 37154,127:Y=PEEK(37152):POKE 37154,255
30 IF X=122 THEN PRINT "UP"
40 IF X=118 THEN PRINT "DOWN"
50 IF X=110 THEN PRINT "LEFT"
60 IF Y=119 THEN PRINT "RIGHT"
70 IF X=94 THEN PRINT "FIRE"
80 GOTO 10
```

Of course, this is only a simple joystick routine, which is probably too simple to be useful in an arcade-style game. To make it a bit more useful, you can use GOTO commands after each THEN. This enables the program to do different tasks based on the position of the joystick. For example, you could add programming to control the cursor's screen position by incrementing or decrementing the cursor's horizontal and vertical position.

Rear-Panel Connections

The VIC-20's rear panel has a lot to offer. Connections for a monitor, floppy disk and cassette-tape drives, printers, cartridges, and even modems are all provided on the back. Commodore did not supply the VIC-20 with industry-standard ports, but you can easily achieve direct connections between Commodore-built peripherals and the VIC-20. Commodore even manufactures interface cartridges, such as the VIC-1011A RS-232C Interface, to convert at least one port to the industry-standard format. This permits devices not made by Commodore, but conforming to the standard, to function with the VIC-20.

The *video port*, the leftmost of the two round sockets seen when facing the rear of the VIC, accepts the cable from the Commodore-supplied RF modulator box, as you have already seen. The video port also allows for direct connection of a monitor, a high-resolution, TV-type display specifically designed for use with computers.

Using a special cable, you can connect a monitor directly to the video port. Such cables can be purchased from local computer stores or you can make your own. (The next section, dealing with the Commodore 64, shows you how.)

Next to the video port is another round socket called the *serial port.* That is where the 1541 Disk Drive or a Commodore printer is plugged in.

If your Commodore system has one or more disk drives in addition to a printer, you'll have to employ a system called *daisychaining* so that all those devices can make use of the computer's single serial port. In daisychaining, your printer's cable is plugged into one of the serial ports on the disk drive's rear panel. If two disk drives are present, the first drive's cable is plugged into one of the second drive's serial ports, and the second drive is actually connected to the computer's serial port. In effect, the printer and disk drives are chained together.

The VIC-20's rear panel also sports three rectangular ports. The largest of these is the *expansion port.* The female expansion-port edge connector accepts any of the VIC-20's memory-expansion cartridges. Various plug-in software cartridges, as well as the VIC-1211A Super Expander, also use this port.

Be sure any cartridge is fully inserted into its appropriate slot before you turn the power on. Inserting or extracting a cartridge while the machine is on can ruin the cartridge or damage the computer.

With plug-in software cartridges, no extra programming is necessary to make the unit work. In effect, the ROM (read only memory) chip(s) within the cartridge become part of the computer. As soon as the cartridge is in place and your computer is turned on, you'll find the program already running. All you need to do is follow the instructions included with the cartridge.

Cartridges like the VIC-1111 16K Memory Expander provide extra RAM (random access memory) work space. The VIC-20's Super Expander, in addition to supplying extra RAM, provides ROM programming, which enhances Commodore BASIC with special graphics and music commands. As you can see, these devices do nothing on their own. But the extra memory they provide, and the Super Expander's language enhancements, permit you to create longer, more complex programs.

The VIC-20's smallest rectangular port, just to the right of the serial port as you face the rear panel, is called the *cassette port.* It is the slot into which the cable from the 1530 Datassette tape program recorder is plugged.

The Datassette is no ordinary tape recorder, although it does use standard cassette tapes—ordinary tape recorders cannot normally be used to save or load VIC programs.

The third port on the VIC-20's rear panel is the *user port.* Located at the rightmost edge of the VIC-20, as you face the rear panel, the user port provides a pseudo-RS-232 interface.

RS-232 is the industry-standard connection for many modems, printers, and even for exchange of data between computers. Commodore's own modems, the 1600 VICMODEM and the 1650 AUTOMODEM, plug directly into the user port. But because the user port does not provide standard RS-232 voltage levels, a slight adaptation is required to give the VIC-20 true RS-232 characteristics and allow it to interface with any RS-232 device. Commodore's VIC-1011A RS-232C Interface makes the appropriate voltage corrections for standard RS-232 operation and truly makes the user port a portal to the rest of the computing world.

The Commodore 64

Hooking Up Your System

The Commodore 64 is a more sophisticated computer than the VIC-20. As such, it offers a wider variety of system configurations. For example, both monitor and TV video output ports are provided; all you have to do is select the appropriate cable and port.

Closely linked to the video decision is the choice of a destination for the computer's audio output. Sound signals can be sent to the television or monitor with acceptable results. However, the Commodore 64 is renowned for its musical capabilities, and by sending the audio signal through your stereo system you can fully enjoy your 64's sounds.

The 64 comes with a video cable designed for linking the computer with the TV switch box. Attach the TV switch box using the same procedure described for the VIC-20. Remove the antenna leads from the screws marked VHF on the rear of your television. Place one lead from the TV switch box under each VHF screw and tighten the screws. Select the COMPUTER setting on the TV switch box by sliding the black lever toward COMPUTER.

Then, to complete the setup procedure, plug one end of the video cable into the TV switch box and the other end into the TV connector jack on the rear of the Commodore 64. The jack is just to the right of the channel selector switch, as you face the rear of the computer. Finally, choose either channel 3 or 4 (pick the one that is not active in your area) on the rear panel of the computer and on your television set.

Unlike the VIC-20, the Commodore 64 has no need for an external RF modulator. Video signals are generated using an internal RF modulator. The Commodore 64's unique design provides video information at both the TV connector port and at the audio/video connector port. The only difference is that the TV connector's signal has been processed for use on an ordinary TV, while the signal available from the audio/video connector has not.

Even though you have an audio/video connector port, Commodore does not supply a monitor cable with the Commodore 64. However, cables that connect either the VIC or the 64 with a color composite monitor can be obtained from outside manufacturers or from local computer stores. A cable is included with the Commodore 1701 or 1702 Monitor.

To attach your Commodore 64 to a color composite video monitor, such as the Amdek Color-I or the Commodore 1701 or 1702, first insert the five- or eight-pin DIN plug into the audio/video connector (it will only go in one way). Then, plug the cable that carries the computer's video signal into the video-in socket of your monitor. Finally, plug the audio-carrying cable into either the audio-in socket of your monitor or into a similar audio-in socket (after using the appropriate adapter) of your hi-fi system.

After interfacing your 64 with some video device, connect the computer to the power supply. Be sure the Commodore 64's on/off switch on the right side panel is set to off; then, insert the power supply plug into the round power socket next to the on/off switch. Finally, plug the power supply into a household electrical outlet.

Before applying any power to your system, be sure that each peripheral is plugged into its appropriate slot on the computer. This prevents damage which might be caused if you tried to plug cartridges or other peripherals into a "hot" computer. Also, if possible, plug the 64 and your TV or monitor's power-supply cords into separate household electrical outlets.

This practice eliminates an interference loop that can disturb the screen image when both devices are plugged into the same household outlet.

Turn on your monitor or television first, then your disk drive, then the printer, then any other peripherals, and finally the Commodore 64. Remember, when you have finished computing, turn everything off in the *reverse* order from that listed above. Also, unplug the Commodore 64's power supply from the household electrical outlet when the computer is not in use. Even though power to the computer is stopped at the on/off switch, power continues to enter the power supply and can cause it to get quite warm.

Besides the power socket and on/off switch, the Commodore 64's right side panel has two game ports. Labeled control port 1 and control port 2, these ports allow the Commodore 64 to use joysticks, game paddles, or a light pen. Joysticks and paddles can be plugged into either one or both ports. However, a light pen (which must be purchased from an outside manufacturer, as Commodore does not produce one) can only be used in control port 1.

Reading a joystick position on the 64 is a much simpler process than it is with the VIC-20. The four stick positions (up, down, left, and right) are assigned numbers (0, 1, 2, and 3 respectively). The fire button is switch number 4. When you press the joystick in one of these four directions, a single internal switch is closed and this information is sent through the joystick's cable to the attached control port. When the joystick is pressed in a diagonal direction, two switches are depressed and their information is sent to the control port.

With the Commodore 64, you have to PEEK only a single memory location to read the joystick's position and fire status. When using a joystick in game port 1, this address is 56321, while a joystick in game port 2 is read from location 56320.

Here is a simple program which will read a joystick in game port 2. It reads the fire button only when the stick is in the center position.

```
1 X=PEEK (56320)
2 IF X=126 THEN PRINT "UP"
3 IF X=125 THEN PRINT "DOWN"
4 IF X=123 THEN PRINT "LEFT"
```

```
5 IF X=119 THEN PRINT "RIGHT"
6 IF X=118 THEN PRINT "UP&RIGHT"
7 IF X=117 THEN PRINT "DOWN&RIGHT"
8 IF X=122 THEN PRINT "UP&LEFT"
9 IF X=121 THEN PRINT "DOWN&LEFT"
10 IF X=111 THEN PRINT "FIRE"
11 GOTO 1
```

Naturally, these IF-THEN statements may not be suited for your particular application. You could include another action after each THEN statement, which could be anything from updating the cursor's screen position to selecting an item from a menu.

The 64 back panel offers several ports. One of them, the *serial port,* is flanked by the audio/video connector port on one side and the cassette interface port on the other. One of the most interesting features of this port is its ability to create a *bus,* which permits up to five serial devices to be handled from this single port. The two most common serial devices are the Commodore disk drive and Commodore printer (or printer/plotter).

Linking one serial device to another is known as *daisychaining* and can be accomplished via a second serial port on each device. When this arrangement is used, the final device is the only one actually attached to the 64's serial port. However, as long as each device on the bus is given its own device number, the computer is able to transmit or receive data, or to give instructions to any device on the bus.

Another prominent port is the *cartridge slot,* on the far left edge when viewing the 64's rear panel; it provides a direct path to the computer's main circuitry. Commodore manufactures application programs and game cartridges that contain their software on ROM (read only memory) chips, which plug into this port and become an integral part of your computer.

Just to the right of the serial port is the *cassette interface port.* Although the 1541 Disk Drive is the most commonly chosen and best-suited mass-storage device made by Commodore for the 64, cost factors make the 1530 Datassette an appealing substitute. The cassette interface port is identical to the cassette port of the VIC-20, so Commodore's 1530 Datassette is compatible with both computers.

Finally, the Commodore 64 provides a *user port.* Located on the right edge of the computer's rear panel, the user port gives the user a programmable interface with the outside world, including RS-232 standard communications. RS-232C is the standard interface between computers and modems, printers, or even other computers. However, the Commodore 64 user port strays from the strict definition of RS-232C in that its voltage range is from 0 to +5 volts rather than the usual −12 to +12 volts. But that can easily be corrected by using a VIC-1011A RS-232C Interface. With this cartridge in place, any RS-232C-compatible device will work with your system.

The Keyboard

Your Commodore computer can receive input from a variety of sources. But the most common and versatile input device is still the keyboard. The alphabetic and numeric keys of the Commodore keyboard are arranged in the usual *qwerty* (the standard typewriter key layout) positions, but some of the computer's special-purpose keys require further clarification. Most of these special keys can be found to the far left or right of the main alphanumeric keys; all are described in Table 1-1.

In using these (and other) keys, you may occasionally get error messages. Commodore 64 error messages are summarized in Table 1-2.

Table 1-1. Commodore Special-Purpose Keys

Key	Use
↑ (Up Arrow)	Exponentiation
← (Left Arrow)	Prints a left arrow
Commodore key	Selects uppercase/graphics or upper/lowercase mode
CLR/HOME (Clear/Home)	Homes the cursor, or (when shifted) clears the screen
CRSR Down/Up	Moves the cursor down, or (when shifted) up
CRSR Right/Left	Moves the cursor right, or (when shifted) left
CTRL (Control)	Used to change the character color or select reverse graphics mode
f1–f8 (Function Keys)	User programmable keys
INST/DEL (Insert/Delete)	Backspacing eraser, or (when shifted) inserter

RESTORE	Restores the computer to its original state when used with RUN/STOP key
RETURN	Enters a program line or a direct command
RUN/STOP	Stops a program's execution, or (when shifted) loads a program from tape
RVS ON (Reverse On)	Selects reverse graphics mode when used with the CTRL key
RVS OFF (Reverse Off)	Returns to normal graphics mode when used with the CTRL key

Table 1-2. Commodore 64 Error Messages

Message	Possible Cause	Possible Solution
BAD DATA	Wrong variable type	Use numeric variable
BAD SUBSCRIPT	Not an element of an array	Use an element within range or increase range
CAN'T CONTINUE	Program error or LIST error	Run or LIST from start
DEVICE NOT PRESENT	Peripheral is not correctly attached	Connect or turn on peripheral
DIVISION BY ZERO	Number divided by zero	Remove zero division
EXTRA IGNORED	INPUT too long	Shorten or remove separators (like commas)
FILE NOT FOUND	Data file not found	File on another tape or disk, or wrong name given
FILE NOT OPEN	Channel not open to peripheral	OPEN file
FILE OPEN	File already open	CLOSE previously OPENed file
FORMULA TOO COMPLEX	String evaluation too complex	Break operations into smaller parts
ILLEGAL DIRECT	INPUT statement used in direct mode	Use only within a program
ILLEGAL QUANTITY	Number out of 64's range	Use number within range
LOAD	Bad cassette tape program	Try LOAD again

NEXT WITHOUT FOR	Loop error	Include FOR statement
NOT INPUT FILE	Data sought from output file	Get data from input file
NOT OUTPUT FILE	Data sent to input file	Write data to output file
OUT OF DATA	No more DATA for READ statement	Include necessary DATA, or limit READ statement
OUT OF MEMORY	Program too long	Streamline program
OVERFLOW	Calculation result too long	Simplify calculation
REDIM'D ARRAY	Variable array reDIMensioned	DIMension each array only once
REDO FROM START	INPUT statement received character data	Supply numeric data instead
RETURN WITHOUT GOSUB	RETURN out of place or not necessary	Relocate or remove RETURN statement
STRING TOO LONG	More than 255 elements in a string	Remove extra string elements
SYNTAX	Statement or command not known by computer	Look for misspelled word or wrong punctuation
TYPE MISMATCH	Wrong data type received	Use numeric or string data as expected
UNDEF'D FUNCTION	User function not yet defined	Define function first
UNDEF'D STATEMENT	Nonexistent line number called	Alter destination of GOTO or GOSUB statement
VERIFY	Faulty SAVE of program	Resave program

Chapter 2

The 1530 Datassette

Chapter 2

The 1530 Datassette

Owning a computer without a tape- or disk-storage system, then, is like owning a car without wheels: You can run the engine, but you can't go very far. One of the most commonly known, easily obtainable, and widely accepted ways of recording electronic signals (like those generated by computers) is on cassette tape.

Not too long ago, in fact, mainframe business and scientific computers required gigantic reels of one-half-inch-wide tape, similar to the tape in ordinary audio cassettes, for storage of programs and data. You can imagine the storage room required to store a few hundred reels of this tape (not to mention the time required to run through an entire tape looking for one piece of information)! To a large extent, this problem was solved with the advent of the floppy disk, and although the floppy disk may seem like the ultimate data-storage medium, tape still retains some advantages. For even though data must be read sequentially off the tape (that is, in one straight line), cassette tapes are familiar to (and therefore more readily acceptable by) almost everyone. They are also relatively inexpensive.

Many personal computers will accept any type of monophonic tape recorder as a peripheral to store programs and data. However, most computer companies have developed and marketed their own tape recording peripherals. Often these devices not only enhance the computer's signal for saving a program on a cassette tape, but also offer direct connection to one of the computer's special I/O ports.

What Can the 1530 Datassette Do?

Because you're probably already familiar with the operation of a standard cassette deck, learning to use the Datassette will consist mainly of learning a few computer commands.

The 1530 Datassette's functions are really threefold: saving programs, loading programs, and manipulating files. The first of these, saving a program, is of immense importance. When you've spent several hours at a keyboard, squinting at a small-print program listing, testing and retesting subroutines, such a saving system is a necessity. The alternative is to type everything back in from scratch (hardly a productive use of your time).

The second important function of the Datassette is loading a previously recorded program into your VIC-20 or Commodore 64. This lets you enjoy the benefits of other people's programming skills from the first day you have your computer system. Commodore itself has many such programs on the market, ranging from computer-related and general application programs to entertainment and educational software. Cassette-based software by other companies covers a wide range of topics and is available both at computer stores that sell Commodore computers and by mail order.

Finally, you can use cassettes and your Datassette to create a data base, that is, storage of files of information, which can be called and used from a data-managing program. For example, if you have a program to keep track of all the names, addresses, and telephone numbers of your friends, you would load this program into your computer first. The program itself contains none of the actual statistical information, but it knows how to get such data off the tape and present it to you. Thus, it circumvents the problem of your computer's limited memory. If all the name/address information had to go into the machine at once, you would be limited as to the number of entries your system could hold. When the file-accessing system is used, however, you have a theoretically infinite library of cassettes for file storage.

The 1530 Datassette

The Commodore 1530 Datassette could easily be mistaken for a common tape recorder. But one obvious difference is sure to catch your eye: there is no external speaker. A computer program tape (as you know if you've ever listened to one) produces nothing more than a high-pitched wavering whine. Such a sound can be extremely grating on the ear; thus a speaker and volume control were not included in the design of the Datassette.

In fact, the Datassette was developed with another listener in mind: the computer itself. The tape's signals are read as usual, but instead of being directed toward a speaker, they are sent through the cable that extends from the Datassette's rear panel to the computer's cassette port. This is the path along which the computer's signals will travel when you save or load a program.

On the Datassette's top side is the compartment that holds a cassette tape. A supply spindle (on the left) and a take-up spindle (on the right) pull the tape through the Datassette. You can see the spindles through the lid of this compartment. The two spindles are controlled by the internal motor, which is in turn controlled by the six push buttons, which are on the recorder's top front edge.

The first button on the right (EJECT) opens the cassette compartment's lid. This permits insertion of a cassette tape. Next to the EJECT button is the STOP button, which stops the tape's movement. Pressing STOP immediately releases any other buttons that might be depressed. F.FWD (Fast Forward) and REWIND are helpful for moving quickly to a specific place on a tape or for rewinding tapes that are to be stored away.

The leftmost buttons of the Datassette are PLAY and RECORD. PLAY makes the Datassette read signals from the cassette and send them to the VIC-20 or Commodore 64. However, when the RECORD and PLAY buttons are pressed simultaneously, any previously saved material is erased from the tape, and the new signals coming from the computer are recorded instead.

When recording is taking place, the SAVE indicator light (a small red LED above and to the right of the row of tape function buttons) is illuminated. Directly above the SAVE indicator light is the counter, a row of three rotating digits and a reset button. Whenever a cassette tape is in motion within the Datassette, the tape-counter digits rotate to keep track of your position on the tape. If you always use the reset button when inserting and rewinding a tape, you can note the position of a program on the tape and refer to it the next time you want that program.

Finally, you will note that there is no power cord on the Datassette; neither is there a compartment for batteries. Instead, the Datassette draws its power from the computer itself. Thus, nothing but the purely mechanical EJECT and tape-counter reset buttons will work unless the Datassette is plugged into a Commodore computer and the computer's power is turned on.

Using the 1530 Datassette

As with the use of any peripheral device, the Commodore computer's power must be off before the peripheral is plugged into an I/O port. The 1530 Datassette's connector will go in only one way. Once the connector has been pushed into place, you can turn the computer system on. Remember to turn on the computer last.

Operation of the Datassette follows a fairly straightforward procedure. Regardless of whether you plan to load or save programs or data, you must first insert the appropriate cassette tape into the Datassette. Examine the cassette tape to determine which side is to be used. Then follow these steps to insert the cassette into the Datassette:

1. Press the Datassette's EJECT button.
2. Remove the cassette tape from its protective case.
3. Orient the cassette tape so that the side to be read from or written to is facing up, and so that the exposed portion of the tape is toward you.
4. Insert the cassette tape along the guides of the underside of the tape compartment lid.
5. Gently press the tape compartment lid down until it clicks into place.
6. Press the REWIND button, to be sure that the tape is wound back to the beginning. You should be able to see the supply and take-up spindles spinning rapidly during this process. As soon as the spindles stop spinning, press the Datassette's STOP button.
7. Press the tape-counter reset button; the digits should then read 000.

When inserting a cassette tape, be sure it is securely seated along the compartment lid guides and also behind the two small protections underneath the lid's front edge. Repeated careless closing of the lid when the cassette is not inserted properly can result in a misalignment of the Datassette's read/write heads. When this happens, programs may no longer load or save properly, and the Datassette must be repaired.

Follow these steps when loading a program:

1. Insert the program tape as described above.
2. Type LOAD and press RETURN.
3. Follow the computer's instructions for program loading as they are presented on your TV or monitor screen.

4. As prompted, press the Datassette's PLAY button until it locks into place.
5. The tape will continue to advance while the program is loading. Once the program is loaded, the tape will stop moving even though the PLAY button is still depressed. Press the STOP button to release the PLAY button.
6. Rewind the tape, as described previously, ending with a press of the Datassette's STOP button.
7. Press the EJECT button to remove the tape.
8. Remove the cassette and return it to its protective case.

When loading, saving, or verifying a program with the Commodore 64, the screen will go blank during the actual operation. The image will return to issue prompts and when the operation's execution is complete.

Saving a program follows nearly the same process as loading a program. With the program to be saved already in computer memory, follow these steps:

1. Remove a blank cassette tape from its protective casing, insert it into the Datassette, rewind the tape, and reset the tape counter.
2. Type SAVE and press RETURN.
3. Follow the instructions for program saving as they are presented on your TV or monitor screen.
4. As prompted, press the PLAY and RECORD buttons simultaneously until both click into place.
5. Check to see that the Datassette's SAVE light comes on. If this light does not come on, check to see that the RECORD button is depressed. If RECORD is not depressed, you'll have to start over again by stopping the computer's SAVE attempt (with the RUN/STOP key), stopping and rewinding the tape, and performing the whole SAVE process again.
6. When the tape stops moving and the SAVE light goes out, the SAVE is complete. Press the Datassette's STOP button.
7. You now have the options of saving the program onto tape again (directly after the first copy, as a backup), rewinding the tape and verifying the saved program, or rewinding the tape and removing it from the Datassette, trusting that it has been saved correctly.

If you choose to use the VERIFY feature, the information that was recorded on the tape is compared against the original program in the computer's memory. Using the computer's VERIFY feature elicits the same Datassette prompts as loading a program does. At the end of the examination, you are told whether the saved program is correct or not. If the VERIFY check shows the tape copy to be faulty, the program should be saved again, copying over the previous copy or moving to a different area on the tape. The problem may lie in an imperfection on the tape's surface. Always make several copies of any saved program. Using a few extra inches of tape is cheaper than retyping (or reconstucting!) an entire program.

As a further precaution, it is possible to alter the cassette so that it cannot be recorded over accidentally. Two small tabs on the back edge of the cassette can be removed to "write protect" the tape. As with normal tape recorders, the Datassette has a sensing probe that tests for the presence of these tabs; when tabs are present, recording can go on normally. But when the tabs are missing, the RECORD button cannot be depressed. If you decide later on that you want to record over the programs on your protected tape, a small piece of masking tape over each of the protect notches will act as surrogate tabs.

It is possible to protect only one side of your tape, if you so desire. Turn the tape so that the side you want to protect is up, while the exposed tape is facing you. Then, remove only the tab which is farthest to your left.

It is not necessary to purchase high-quality specialty or metal cassette tapes for program recording. In fact, they are not recommended. Average quality 30-minute (C-30) or 60-minute (C-60) cassette tapes are generally reliable (and also cheaper), and they allow several copies of many programs to be contained on a single tape.

Program Applications with the 1530 Datassette

The most common Datassette operations—loading, saving, and verifying programs—are done outside a program format by using direct commands. Basically, all you have to know are the commands LOAD, SAVE, and VERIFY. When you use these commands, the Commodore computer will prompt you on the steps you should take.

LOAD. A generalized syntax for the LOAD command would be:

LOAD "PROGRAM NAME"

When a specific program is to be loaded into the Commodore computer, the program's name must be included within quotation marks after the command LOAD, if this program is not the first one on the tape. The screen will announce the name of the discovered program and then will load it or continue its search for the desired program. There is a short pause during the announcement before the search continues or the program is actually loaded, so that you have time to read the screen. To shorten this waiting time, you can press the space bar. Otherwise, you can wait for the loading or searching to continue on its own.

Once the program has been loaded, you will still have to give the command to make the program run. However, there is an alternative loading method which causes the program to be executed as soon as it has been loaded. Because you will be unable to specify the name of the program you wish to load, the desired program will have to be next on the cassette tape. Instead of typing the word LOAD, hold the SHIFT key and simultaneously press RUN/STOP. The standard LOAD prompts take place and you should follow them as usual. However, you will not see the final READY indicator on the screen. The program begins running instead. You should, as usual, STOP and REWIND the tape in the Datassette, because it continues to draw power from the computer.

SAVE. Saving a program is just as straightforward as the LOAD operation. This is a typical SAVE command line:

SAVE "PROGRAM NAME"

After you have followed the computer's instructions and pressed the Datassette's RECORD and PLAY buttons, your part of the SAVE process is done. The computer controls the Datassette, even stopping its motor when the Datassette has finished saving the program. All you have to do is manually press the Datassette's STOP button to release the RECORD and PLAY buttons. The READY notice and flashing cursor are absent from the screen while a program is being saved, but they will reappear when the process is complete.

If a program is being saved at the very beginning of a cassette tape, you may be worried that the entire program will not be recorded due to the cassette's leader tape. But there is no cause for concern. The computer does not begin saving the program immediately. Instead, during the first five revolutions of the tape-counter digits, a steady tone is generated to prepare the Datassette to record the tones of a computer program. Usually, the leader is short enough so that the program recording begins on the recording tape after the leader. If you watch the SAVE indicator light from the beginning of a SAVE through its completion, you should see this light dim briefly at the fifth tape-counter digit revolution.

VERIFY. After saving a program, but before it has been erased from the computer's memory, you can check to see that it was saved properly. Commodore's VERIFY command issues the same type of instructions to the user as the LOAD command does. When the VERIFY command is followed by the name of a program, the computer searches the titles of the programs on the current tape until it finds a matching name. The syntax for the VERIFY command line, when specifying a program's name, follows:

VERIFY "PROGRAM NAME"

However, it is not necessary to specify the name of a program to be verified. When given alone, the VERIFY command will examine the next program on the cassette tape. In either case, once the computer has finished its comparison, the outcome is shown on the screen. A positive match results in an OK message. But when a discrepancy is noted between the programs, the screen reads ?VERIFY ERROR.

A VERIFY ERROR message can mean that a saved program is faulty (possibly due to an imperfection on the tape's surface) or, when verifying an unnamed program, that you have tried to compare two entirely different programs. Try verifying the program once more. If the error occurs again and you're sure that you are verifying the correct program, find a different blank area of the tape and try saving your program again. Then VERIFY this newly saved program.

Because of the sometimes finicky behavior of cassette tapes, it's always a good idea to take precautions to keep a program safe once it has been recorded. The tape should be kept away from strong magnetic fields, such as those found near a television set or audio speakers. Prolonged exposure to such fields can erase programs from tape.

Files

The Datassette continues to be useful even after a program is loaded, saved, or verified. Using a few simple programming lines, you can also store files of data on a cassette tape. Later this data can be called back from within a program.

Any file must first be established with the use of the OPEN command. At that time, the file is given a name. For example, OPEN 4,1,1,"FILENAME" will open a file on the tape and label it with its name.

The three numbers after the OPEN command are really not difficult to understand. The first number, 4, is the *file* number. There is no special significance attached to the number 4; actually, the file number can have any value from 0 to 255. However, when performing other operations with the same established file, use the same file number for operations that require the file number in their syntax.

The second number, 1, is the *device* number. The number 1 must always be used as the device number when performing Datassette operations. Because the same file commands used with the Datassette are also used with Commodore's disk drive, the destination for the data is controlled by the device number.

Finally, the third number (in this example, 1) specifies what type of data exchange will go on with the opened file. This *secondary address* number, as it is called, can have a value of 0, 1, or 2. When a file is opened with 0 as the secondary address, the computer is prepared to read data from that file. When the value is 1, data will be written on the tape and a marker is placed after all the data so that the end of the file will be known. This marker is called an end-of-file marker. When a file of secondary address 2 is written, an end-of-tape marker is written onto the tape. Encountering this marker makes the computer think that the end of the entire tape has been reached and that no further data exists.

This is the general syntax for an OPEN command line:

OPEN file number, device number, secondary address, "FILENAME"

Writing data onto a cassette tape is, in some ways, analogous to PRINTing words on the TV or monitor's screen. In fact, a statement called PRINT# is used to place the data on a tape. Once a Datassette file has been opened (using 1 as the secondary address to specify a writing operation), the PRINT# statement is followed by the file's file number, a comma, and the data to be stored. This data is often in the form of a variable (for example, PRINT#9,P$—where 9 is the previously established file number and P$ is a string variable that stores the data).

Application of the PRINT# statement will elicit screen prompts that give instructions in using the Datassette to save the data. Remember that there are two ways of marking stored file information on a tape, and both methods are dependent on the way that the file has been opened. For most operations the file should be opened with a secondary address of 1 so that an end-of-file marker will be written onto the tape when the file is closed. The end-of-tape marker written at the CLOSE of a secondary address 2 file, when encountered, causes the computer to notify you that no more information is available. In fact, you will be informed that the Datassette device is not even present. Obviously, the secondary address 2 should be reserved for special situations.

Generally, the format to follow for writing data onto a tape is PRINT# file number, data. You can devise the system for giving the data variable its value to best suit your specific programming application.

After any file operation has been completed, the opened file must be closed with the CLOSE statement. For instance, CLOSE 3 closes the file opened with a file number of 3. The simple syntax for the CLOSE statement is as follows:

CLOSE file number

It is always good practice to OPEN a file immediately before use and to CLOSE it immediately after its use is complete. This prevents data from being written to or read from a file which was accidentally left open. Errors can also occur when you try to OPEN a file that is already open or to CLOSE a closed file.

When you want to retrieve the information stored on tape, OPEN the file (using a secondary address of 0) and specify the file's name. The GET# statement does the actual job of retrieving the data. For example,

GET#7,Q$

will receive information from a file with file number 7 and store the information in a string variable called Q$. Any kind of data (letters, numbers, or punctuation) can be retrieved in this manner. This is the general syntax for the GET# statement:

GET# file number, string variable

Additional programming could include a method for repeatedly retrieving pieces of information. Once again, the file should be closed when you have finished accessing the data.

Occasionally, you may get the error message FILE DATA while using the Datassette. It means that you are using the wrong variable type. To solve the problem, use a string variable.

Reference Section

Syntax of Datassette Functions

Function	Command
Close a file	CLOSE file number
Load a program	LOAD "program name"
Open a file	OPEN file number, device number, secondary address, "filename"
Receive data	GET# file number, string variable*
Save a program	SAVE "program name"
Transmit data	PRINT# file number, data*
Verify a program	VERIFY "program name"

* An OPEN command is required prior to these functions; a CLOSE is required afterward. *Do not include the asterisk (*) as part of the command.*

Glossary

Data base. An electronic file of information, stored on tape or disk for use (usually) by another computer program. The same term applies to the program that creates these files.

Leader tape. The strip of nonsensitized tape that connects the actual recording tape with the cassette's reels.

Read/write head. The unit that loads from (or saves to) the cassette's magnetic tape.

Tape-erase protect notch. The notch that appears when the cassette's edge tabs are removed; used to protect a cassette tape from being erased or recorded over.

Chapter 3

The 1541 Disk Drive

Chapter 3
The 1541 Disk Drive

Whether you're a programmer or not, mass storage of information is still a major concern. The ideal memory device would possess great speed in both transmission and reception of data, have a large capacity, and be highly reliable. For the most part, those specifications describe the magnetic floppy disk.

What Can the 1541 Disk Drive Do?

With the 1541 Disk Drive, the Commodore computer becomes a powerful information-handling system. Numeric and text data can be stored in both file and program form—and the capacity of a floppy disk far exceeds that of a standard computer cassette tape.

The 1541 can store 144 directory entries, or a little over 170K of information, on a single floppy disk (this includes both files and programs). However, with various types of file storage the total floppy-disk capacity drops to approximately 163K.

Speed is also an important feature of the 1541 Disk Drive. For example, a program of roughly 8K can be saved in less than 15 seconds, and there is nothing comparable in the Commodore line of peripherals when it comes to fast storage and retrieval of files. Thus, it is not surprising that, for many Commodore users, the disk drive is an essential part of the system.

A Tour of the 1541 Disk Drive

The 1541 is a self-contained system. All of the components for controlling and operating the 1541 are enclosed in its housing. There is a disk controller, disk operating system (DOS), 16K of ROM, and 2K of RAM. This makes the 1541 Disk Drive a *smart* peripheral. In other words, all the work is performed by the 1541 without sacrificing any of the computer's memory. This method reduces processing time and leaves the computer free to pursue other activities.

On the front panel is the slot for insertion of a floppy disk. The front panel also has two indicator lights. The green one signals that power is on, while the red one indicates data flow or any drive errors. On the rear panel are the on/off switch, the receptacle for the power cord, a fuse holder, and two Commodore serial ports. These Commodore serial ports are of special interest, since they enable the 1541 to be daisychained either to another disk drive or to a printer.

Using the 1541

The placement of most peripherals is immaterial. Items like a joystick can be placed almost anywhere, but a disk drive is a different matter. An ideal position for the disk drive is within arm's reach of the keyboard, but set so that it will not interfere with either use of the keyboard or other peripherals.

The disk drive is easy to connect. First, make sure that both the 1541 and the computer power switches are off. Only two connections are necessary on the 1541 Disk Drive. The first, and most obvious, is the power-cable connection. The gray cord supplied with the drive is the power cable; it should be fitted into the power-cord receptacle on the drive's rear panel.

The other cord supplied with the disk drive is the Commodore serial connecting cable. This provides a communications link between the computer and the disk drive. Both ends of the cable are exactly the same. Therefore, either end can be plugged into the computer and/or disk drive. The plugs are keyed so that they will only go in one way.

Note the extra Commodore serial port on the rear panel of the 1541. It allows the connection of either another disk drive or a printer. This process of back-to-back connecting is known as daisychaining. In fact, daisychaining lends greater flexibility to the Commodore computer system, allowing a disk drive and a printer (or another disk drive) to occupy the same serial port on the computer. One limitation of daisychaining is that a maximum of four disk drives plus one printer (sometimes two) can be used at the same time.

With all the cable connections complete, you can plug the main power cord into a household socket. Make sure that no disks are in the drive and that the packing material has been removed from the disk slot. Now, turn the drive on, and a slight whirring sound should be heard. Also, the green indicator light on the front panel should come on. The red indicator light will also come on briefly. After this visual and audible confirmation, you can turn on the computer and begin normal computing activities. If there is a problem, turn off the disk drive and recheck all connections. Should the problem persist, consult your dealer.

Differences between the VIC-20 and Commodore 64 require that different data transfer speeds be used for each. In other words, the 1541 must be set at a different speed when working with a VIC than it is when working with a 64. When the 1541 is turned on, it defaults to the 64's data speed, so you should always change the speed after power on if you are using a VIC. The speed change is accomplished through a simple direct command, which should be entered immediately after all the computer components have been turned on:

```
OPEN 1,8,15,"UI-": CLOSE 1
```

Remember to press the RETURN key (this will be assumed from now on).

No alterations are necessary when using the 1541 with a Commodore 64. If, for some strange reason, the drive is inadvertently set to VIC-20 speed when you plan to use it with a 64, the following direct command line will change it back to 64 speed.

```
OPEN 1,8,15,"UI+": CLOSE 1
```

Disks

If you plan to do anything besides run prepackaged software, you'll have to purchase floppy disks. The 1541 requires 5-1/4-inch, single-side, single- (or double-) density floppy disks. Try several brands to find the best disks to use. Like audio tapes, some disk brands do offer greater reliability and better warranty coverage.

Store your disks properly and you'll get years of trouble-free service. Avoid magnetic fields, moisture, and temperature extremes. Keep disks in their protective jackets whenever they are not being used. Several manufacturers make disk storage cases that you might want to use.

Once you have purchased your disks (or your prepackaged software), here's what to do. Check to see if the write-protect notch is covered. The notch is located on the right side of the disk near the top, as you hold the disk with the label at the top. Some software will work properly only with the notch uncovered (which means the disk can have information written to it or *erased from it*). To protect the disk from being erased accidentally, simply cover the notch with a piece of the special tape included with most disks. Remember to remove the protective tape whenever you want to SAVE or SCRATCH something on the disk.

After turning the system on, insert the floppy disk. Make sure that the label side is up and closest to you. In other words, the label should be the last part of the disk to go into the drive, and the writing on the label will be facing away from you. Push the floppy disk in until it clicks into place. Then, close the disk drive door.

After you give a disk command, a whirring noise indicates that the disk drive is functioning. The red light indicates that data is being transmitted or received. This is NOT an error indication; however, if the red front-panel light stays lit after the drive's whirring has stopped, then an error has occurred. Review your actions or address the error channel (see the instructions below) to find the problem.

Take care of your disk drive, and avoid subjecting it to physical shocks that could misalign the read/write head. *The Commodore disk drive is much more susceptible to damage than the Datassette.* You should also give the head a weekly or monthly cleaning (depending upon the frequency of drive usage) with a special head-cleaning disk. Remember that the read/write head in the 1541 Disk Drive is located on the bottom of the drive. Therefore, when cleaning the head, be certain that the cleaning disk is properly inserted. Most head cleaners are designed to work correctly with their label side up, as they are inserted into the drive.

Program Applications with the 1541 Disk Drive

The 1541 Disk Drive can be programmed in two different modes: *direct command* mode and *program* mode. Both can be used for the usual saving and loading of files and programs, but the direct command mode provides exclusive control over the drive's many housekeeping functions.

A housekeeping function can best be described as the preparation and maintenance of the data for a floppy disk. Except when dealing with prepackaged software (where LOAD is the only command that is needed), at least one of these housekeeping functions has to be used before trying any other disk drive command.

Formatting (or NEWing). The most important of these functions is disk formatting. This *must* be done to every new blank disk before it can be used on the 1541. Formatting a disk erases whatever was there and prepares the disk to accept files. *Warning: NEVER format a prepackaged software disk, unless you want to erase the program.*

To format the disk, enter the following line:

OPEN 1,8,15:PRINT# 1,"NEW0:name,number":CLOSE 1

Several bits of information concerning formatting are true for all housekeeping functions. Unless more than one disk drive is used, the 1541 is always referred to as device number 8 in an OPEN command (just as in SAVE and LOAD). Similarly, in the disk commands, the drive number will always be 0. These two facts will change if more than one disk drive is used.

When housekeeping functions are being directed to the disk drive, they must use the command channel. This channel is labeled 15 in the disk-formatting example. Finally, the number 1 in the OPEN, PRINT#, and CLOSE commands above is a user-assignable file number. Thus, a more generalized syntax for disk formatting would be as follows:

OPEN file number, device number, channel number:PRINT# file number,"NEW drive number:name,number":CLOSE file number

The actual disk-format command is NEW, which can be abbreviated as N. Using this shorthand form can save time when formatting several disks. In addition, you should determine your own style in OPENing a communications link with regard to file numbers. Pick an easy-to-remember number and stick with it. That will make the actual memorization of the housekeeping functions easier.

COPY. You can copy files and programs on the same floppy disk by using COPY. This function should not be confused with the backup program sold by Commodore. With COPY, a duplicate of a file or program is saved onto the original disk. In this way, another version of a prized program can be loaded many times without fear of damaging the original. However, if the disk itself is destroyed, *both* the original and the copy will be lost. For this reason, it is always a good idea to keep backup copies of your important programs on separate disks, filed in a different area from your working copies. This way, you can use one copy while maintaining a personal archive of your programs in a safe place.

To see how COPY works, look at the following line. It will COPY "MATH" and call it "MATHNOTES."

OPEN 3,8,15:PRINT#3,"COPY0:MATHNOTES=0:MATH": CLOSE 3

The generalized format for COPY is:

OPEN file number, device number, channel number:PRINT# file number,"COPY drive number:copy name=drive number:original name":CLOSE file number

The abbreviation for COPY, the letter C, may be used in place of the word COPY. Additionally, the COPY function permits the inclusion of several files or programs into one large copied file or program. For instance, you could use something like this:

OPEN 2,8,15:PRINT# 2,"C0:copy filename=0:filename 1,0: filename 2,0:filename 3,0:filename 4":CLOSE 2

A maximum of four files can be combined in this COPY function. Any sequence of filenames is permissible. Just make sure that the syntax is precise. The drive number and a colon (0:) must precede each original filename.

INITIALIZE. The function that will restore the 1541 Disk Drive to its start-up condition is INITIALIZE. Most serious drive hang-ups can be corrected with this function. The only other way a disk-drive error can be fixed is by turning the drive off and then back on. The following line will initialize the 1541 Disk Drive:

OPEN 6,8,15:PRINT# 6,"INITIALIZE":CLOSE 6

Here is the generalized format for INITIALIZE (abbreviated by the letter I):

OPEN file number, device number, channel number:PRINT# file number:PRINT#filenumber,"I":CLOSE file number

After using this function, VIC owners will need to reset the drive to run at VIC-20 speed.

RENAME. Any file or program in which a name change is needed can be handled with RENAME. The syntax is similar to COPY:

OPEN 9,8,15:PRINT# 9,"RENAME0:new name=old name": CLOSE 9

Use this generalized format for RENAME:

OPEN file number, device number, channel number:PRINT# file number,"RENAME drive number:new name=old name": CLOSE file number

Like the other housekeeping functions, RENAME has an abbreviation (the letter R). But keep one thing in mind: RENAME will not operate on files that have not been properly CLOSEd.

SCRATCH. A function perfectly suited to housekeeping, SCRATCH is used to erase undesired files and programs. Following this elimination, the space occupied by the discarded files can be reused. This line will remove the specified file:

OPEN 11,8,15:PRINT# 11,"SCRATCH0:filename":CLOSE 11

The generalized format for SCRATCH is:

OPEN file number, device number, channel number: filename: CLOSE file number

When using SCRATCH (abbreviated by the letter S), make sure that the specified file really is the file that you want removed. Once a file has been SCRATCHED, it is gone from the disk for good, unless you use some pretty tricky techniques to get it back.

VALIDATE. The last housekeeping function is VALIDATE. VALIDATE acts as a maid service, tidying up the disk after you have performed other functions and activities. The following line will VALIDATE a floppy disk:

OPEN 14,8,15:PRINT# 14,"VALIDATE":CLOSE 14

The generalized format for VALIDATE is:

OPEN file number, device number, channel number:PRINT# file number,"VALIDATE":CLOSE file number

VALIDATE can be abbreviated with the letter V. The best time to use it is after performing several SAVEs and SCRATCHes. That way, all the excess space can be concentrated and applied during future program or file storage.

SAVE, LOAD, and VERIFY

Many other commands are available and understood by the 1541. The three most common are SAVE, LOAD, and VERIFY. All should be familiar to any former Datassette user, and each functions exactly as it does with the Datassette, except the syntax is slightly different.

SAVE. To save a program, use the following line:

SAVE"PROGRAM NAME",8

The 8 serves as a device number and represents the disk drive. One feature of SAVE is replacing an earlier program with an updated or corrected version. This is extremely useful when working the bugs out of a custom-created program. To save and replace a program, use the following line:

SAVE"@0:PROGRAM NAME",8

By using the @0: character combination, you tell the drive to erase the previous program version and save the new version in its place. A command number, 1, should be used if required. It should be noted here, though, that this function occasionally malfunctions on some Commodore disk drives. Another way of replacing a file is to save the new version using a new filename, SCRATCH the old version, then save the program again (if you so choose) under the original name. Many programmers find it convenient to use numbered versions of programs in progress (for example, STARGAME1, STARGAME2, and so forth).

LOAD. Use this form to load a program:

LOAD"PROGRAM NAME",8

The LOAD is performed just as with the Datassette, except the program is stored on a floppy disk instead of a cassette tape. As with the SAVE commands, the command number (1) should be used if the program is designed to go into a specific memory location.

A valuable feature of LOAD is its ability to examine a floppy disk's directory. The directory of a floppy disk is like a book's table of contents—it gives you a complete record of every program or filename, as well as a listing of how much space each one occupies and the amount of space remaining on the disk. To load the directory, type the following:

LOAD"$",8

Almost instantly the READY cursor reappears—but that's all that happens. The reason the directory isn't immediately printed on the screen is that it has been loaded into the computer's memory. To view the memory's contents, use the LIST command. Now, the entire directory will scroll upward on the screen.

To make a printout of the directory that has already been loaded properly, make sure the printer is connected and turned on, then type OPEN 1,4:CMD 1:LIST and press RETURN. After the LISTing is complete, type PRINT# 1:CLOSE 1 and press RETURN.

You can also use LOAD without direct mention of the program name. This is accomplished through pattern matching. By using the asterisk (*) and the question mark (?), you can abbreviate program names, account for spelling errors, and perform security LOADs. The ? is a letter-specific substitute, meaning that LOAD"DI?",8 would load programs with such names as DIP and DIM, which are only three letters in length. On the other hand, * serves as a broad-stroke substitute. For example, LOAD"DI*",8 would load a program whose name began with the letters DI and was of any length. Actually, the computer will load the *first* file on the disk with a name that matches the pattern, so if no program has been loaded since the disk drive was initialized, LOAD"*",8 will load the first program on the disk. If a program has been loaded, LOAD"*",8 will load that file.

VERIFY. To VERIFY a program, use this line:

VERIFY"PROGRAM NAME",8

In this case, there is no need for a command number. However, a VERIFY ERROR might occur with the VIC-20 if various memory-expansion devices were used indiscriminately.

When a 1541 error occurs, there is no way of immediately knowing the source of the problem. However, with three lines of programming, the error channel can be read. Since the reading of the error channel can be done only with programming, this would be an ideal candidate for a GOSUB in any program that makes use of the 1541 Disk Drive. A routine for reading the error channel follows:

```
1 OPEN15,8,15
2 INPUT#15,W,X$,Y,Z
3 CLOSE15:PRINTW;X$;Y;Z
```

Variable W represents the error number, X$ is the name of the error, Y is the track on the disk where the error occurred, and Z is the block number where the error occurred.

Sequential Files

As the name implies, sequential files are read and written in order, from their beginning to their end. Although this limits their uses, it does make programming simpler. After a sequential file has been properly OPENed, three commands are available: PRINT#, INPUT#, and GET#. While the first command is for writing data, INPUT# and GET# are for reading data.

OPEN. The syntax for OPENing a sequential file is as follows:

OPEN file number, device number, channel number,"0: program name,file type,read or write"

File type is the first-letter abbreviation for a program (P), sequential file (S), user file (U), or relative file (L). Also, you must indicate whether the file will be read (R) or written (W).

PRINT#. The syntax for PRINT# is:

PRINT# file number, data

The data may be either contained within quotation marks ("") or represented by a string variable. Careful use should be made of punctuation—particularly, commas and semicolons—to avoid problems when the file is read later. At the conclusion of data entry, an end-of-file (EOF) marker will be automatically entered for you.

INPUT#. The syntax for INPUT# is:

INPUT# file number, variable

When reading with INPUT#, a separator is necessary to isolate each of the file entries. This separator may be a comma (,), semicolon (;), or a carriage return (CHR$(13)). But the same precautions exercised with PRINT# must be followed with INPUT# to insure data integrity.

GET#. GET# command's syntax is as follows:

GET# file number, variable

Data is read byte by byte, one character at a time, with GET#. A string variable should be used for data retrieval, because of the way BASIC likes to have data entered. After the data is received in string form, simply convert the numeric data using the BASIC VAL function.

Finally, if you are using more than one disk drive, the device numbers will have to be reassigned. You can accomplish this in two ways: with either a simple program line or a hardware modification. If you choose the hardware method, see your dealer for the proper procedure. (You can also check the 1541 user's manual).

The Program Line Procedure

The program line procedure is fairly easy to master, although some care is required. Device numbers 8–11 inclusive may be assigned to the disk drives. First, connect the main disk drive as described in this chapter. Plug in the power cable only for the other 1541 Disk Drives; then attach the black serial cable to each of the additional drives. DO NOT daisychain them yet.

Next, turn on the entire system, including the additional drives. OPEN the command channel and send the commands like this (replace *device number* with 9, for example, to assign that number to the drive):

```
PRINT# file number,"M-W"CHR$(119)CHR$(0)CHR$(2)CHR$
(device number+32)CHR$(device number+64)
```

CLOSE the channel after changing the device number of the drive. Then, connect the first additional drive to the main 1541. Repeat this procedure until all the drives have been assigned a different device number. Many public domain utility programs (which can legally be copied) are available to help you change the disk device numbers easily; one such program is on the Commodore bonus pack disk.

The program below allows you to change the device number of a drive quickly and easily. Turn off all drives except the one with the device number you want to change. The current device number of an unmodified 1541 Disk Drive is 8. After changing one drive to 9, for example, turn on the next. If you want to change the device number on that drive, simply run the program again. This way, the drives can be connected in advance. The simple trick is that when a drive is off, it can't be programmed. By programming the drives one at a time, you can have several drives operating with the same computer, using device numbers ranging from 8 to 11.

```
10 PRINT"TURN ON ONLY DRIVE TO BE CHANGED."
20 INPUT"CURRENT DEV#";C
30 INPUT"NEW DEV#";N
40 OPEN 15,C,15
50 PRINT#15,"M-W" CHR$(119)CHR$(0)CHR$(2)CHR$(N+32
   )CHR$(N+64)
60 PRINT"IGNORE ERROR. TURN ON NEXT DRIVE."
70 PRINT#15:CLOSE15
```

Reference Section

Syntax of Disk Functions

A generalized syntax is given for each function. An OPEN command should be given before each of these functions and a CLOSE afterward. Disk controller commands not discussed in the text are included for reference in this section.

Function	Command
BLOCK-ALLOCATE	PRINT# file number,"B-A:" drive;track;block
BLOCK-EXECUTE	PRINT# file number,"B-E:" channel number;drive;track;block
BLOCK-FREE	PRINT# file number,"B-F:" drive;track;block
BUFFER-POINTER	PRINT# file number,"B-P:" channel number; location
BLOCK-READ	PRINT# file number,"B-R:" channel number;drive;track;block
BLOCK-WRITE	PRINT# file number,"B-W:" channel number;drive;track;block
COPY a file	PRINT# file number,"C0:copy filename=0:filename"
Format a disk	PRINT# file number,"N0:name, ID number"
INITIALIZE	PRINT# file number,"I"
LOAD a directory	LOAD"$",8*
LOAD a file	LOAD"name",8*
LOAD file where saved	LOAD"name",8,1
MEMORY-EXECUTE	PRINT# file number,"M-E" CHR$(address lo)CHR$(address hi)
MEMORY-READ	PRINT# file number,"M-R" CHR$(address lo)CHR$(address hi)
MEMORY-WRITE	PRINT# file number,"M-W" CHR$(address lo)CHR$(address hi)CHR$(number of characters)data
Random file	OPEN file number, device number, channel number,"#buffer number"
Relative first OPEN	OPEN file number, device, channel number,"filename,L,"+CHR$(length of record)
Relative OPEN	OPEN file number, device, channel number,"name"
Relative position	PRINT# file number,"P"CHR$(channel number + 96)CHR$(record number lo)CHR$(record number hi)CHR$(position)

* No OPEN or CLOSE commands are necessary. *Do not include the asterisk (*) as part of the command.*

RENAME a file	PRINT# file number,"R0:new name=old name"
SAVE a file	SAVE"name",8*
SCRATCH a file	PRINT# file number,"S0:filename"
Sequential file	"0:name,file type,read or write"
Sequential GET#	GET# file number, variable
Sequential INPUT#	INPUT# file number, variable
Sequential PRINT#	PRINT# file number, data
USER commands	PRINT# file number,"Ucommand number"
USER1	PRINT# file number,"U1", channel number;drive;track;block
USER2	PRINT# file number,"U2", channel number;drive;track;block
VALIDATE a disk	PRINT# file number,"V"
VERIFY a file	VERIFY"name", 8*

* No OPEN or CLOSE commands are necessary. *Do not include the asterisk (*) as part of the command.*

Error Messages for the 1541 Disk Drive

Message	Possible Cause	Possible Solution
0	No error	
1	A file has been scratched	
20	Block header not found	Reset disk, check hardware, use a new disk
21	No sync mark	Reset disk, check hardware, use a new disk
22	No data block	Rewrite data
23	Checksum error	Rewrite data
24	Byte error	Rewrite data
25	Verify error	Resave data
26	Protection notch covered	Remove write-protect tab
27	Checksum error	Rewrite data
28	Sync mark error	Reformat disk, check hardware
29	Disk format error	Reformat disk
30	Command syntax error	Rewrite statement
31	Invalid command	Rewrite statement
32	Statement too long	Shorten statement
33	Invalid filename	Check filename
34	No filename	Insert filename, check for stray colon (:)
39	Invalid error channel command	Rewrite statement
50	Reading past the last record	Reposition record

51	Record overflow	Adjust either record size or data input or both
52	Relative file overflow	Restart with new disk
60	File already open	Close previously opened file
61	File not open	Open file
62	File not found	Place correct disk into drive; make sure of filename
63	File with same name already exists	Rename file
64	File type mismatch	Reenter file type
65	Block already occupied	Specify another block
66	Incorrect track or sector	Use another block, reformat disk
67	Illegal track or sector	Check disk format
70	No available channels	Use an existing channel, close a previously opened channel for reuse
71	BAM (block availability map) mismatch	Reinitialize the disk
72	Disk full	Use another disk
73	DOS mismatch	Use 1541 formatted disk for writing
74	Drive not ready	Place a disk in the drive

Glossary

Commodore serial. A nonstandard serial port. Serial ports transmit information a bit at a time.

Daisychaining. Connecting Commodore peripherals to the Commodore computer via the serial port.

DOS. An acronym for *D*isk *O*perating *S*ystem. This is a specialized portion of memory used for controlling the disk drive. On the 1541, this memory is contained within the disk drive unit.

Read. A term applied to the act of receiving data.

Write. A term applied to the act of transmitting data.

Chapter 4

The VIC-1525 and -1526 Printers

Chapter 4

The VIC-1525 and -1526 Printers

The printer can be considered the workhorse of the computer world. Its grandparent, the typewriter, revolutionized written manuscripts, so it is only natural that the printer has become important in the computer world, too. However, while in the past most businesses were satisfied with the standard typewriter, the increasing availability of business-related information from computer-based sources and the computer's graphics capabilities have created the need for printing devices with more features than the ordinary typewriter.

There are two major kinds of computer printers: letter quality and dot-matrix. Letter-quality printers (for example, the daisywheel printer) produce characters from individual type, similar to that of a typewriter. The typeface strikes against a ribbon and leaves a mark on the paper. The name daisywheel is derived from the wheel that holds each character type at the end of an extension, like the petals of a daisy.

Such printers are praised for the print they produce, which looks as good as if it were typed manually. However, although daisywheel printers may be faster than most human typists, they are usually not as fast as dot-matrix printers. Also, letter-quality printers are limited in the characters they can produce. Interchangeable type wheels are available for some models, but the high-resolution graphics of the dot-matrix printer are impossible to re-create with a daisywheel printer.

Just as photos in the morning paper and pictures on television are composed of tiny dots of ink or light, a dot-matrix printer creates a character by leaving tiny dots of ink on the paper. Rather than being limited by a set of fixed character shapes, the set of wire pins on a dot-matrix print head can be activated individually to create any image that the computer can produce, including special graphics characters.

The main problem with dot-matrix printers is that if the dots are not printed closely enough together, the type has a ragged appearance. Often such type is considered to be draft quality, at best, and not acceptable as a finished product. However, some dot-matrix printers have pins that are close enough together to fill in the gaps between dots, and many double-strike each character. This has resulted in print quality that is virtually indistinguishable from daisywheel print.

What Can These Printers Do?

VIC-1525 Graphic Printer. Commodore's VIC-1525 Graphic Printer can be used with either a VIC-20 or a Commodore 64. It is a dot-matrix device that prints characters from left to right (unidirectionally) at 30 characters per second. Because this printer is designed around the tractor-feed method of paper advancement, standard typewriter paper cannot be used. Tractor paper has pin holes along its edges so that the paper can be drawn through a tractor-feed printer. Perforations allow the tractor-feed edges to be removed, leaving a standard-sized printed page.

One feature that makes this printer superior to most other printers for use with the Commodore system is its ability to immediately interface with the Commodore system. This is evident not only in the simplicity of hardware connection, but also in the use of the the printer's features. The VIC-1525 can immediately "talk" with either the VIC-20 or the Commodore 64, using only a few BASIC commands. While some other printers have added features and better print quality, most of them require special interfaces. In short, the Commodore printers are easier to use for the beginner, but they often have fewer features.

Although many dot-matrix printers can be made compatible with Commodore computers by using special interfaces, there is one function that these printers can rarely perform. The Commodore character sets in both the VIC and the 64 include special graphics characters. The 1525 will reproduce these characters just as easily as any alphanumeric character. Many other dot-matrix printers do not support these unique characters. Recently, interface manufacturers have taken note of this drawback, so that some interfaces now include special memory chips that support printing Commodore graphics.

In addition to printing all of the Commodore characters, the 1525 can also be programmed to print custom-designed characters. If you want to print a character that is not in the Commodore set, a few program lines can give you the desired shape. For example, double-width and reverse-field printing are among the possible style choices. As a matter of fact, because of the 1525's versatility, you can even copy the contents of the screen to the printer.

Besides its uses as a program lister and graphics design producer, the Graphic Printer's 6 × 7 dot character size makes it suitable for note taking. With the appropriate word processing software or self-constructed programming, the Graphic Printer could even be used for casual letter writing or draft-quality manuscript production.

VIC-1526 Dot-Matrix Printer. Despite obvious cosmetic differences, Commodore's 1526 Dot-Matrix Printer is very similar to the VIC-1525 Graphic Printer. Although both are capable of producing the entire Commodore character set (including special graphics symbols), there are a few differences in the way they print. For example, the 1526 can print bidirectionally, making its average speed (about 50 characters per second) faster than that of the 1525.

An 8 × 8 dot character matrix gives 1526 copy a more professional look than that obtained from the 1525. True descenders on the letters *g*, *j*, and *y* also make the print look better. In addition, the larger character matrix permits the creation of even more detailed custom characters.

A microprocessor within the 1526 controls printer operations. As a result, you can even store formatting data in the printer by sending special commands.

A Tour of the Printers

VIC-1525. Since it is larger and heavier than the VIC-20 or the Commodore 64, the VIC-1525 Graphic Printer is an imposing peripheral. Its rear panel contains the connections and controls necessary to prepare the printer for use. As you face this panel, the on/off switch will be on the right edge. Immediately to the left of this switch is the permanently attached power cord (for connecting the printer to a household electrical outlet). The fuse holder is located on the left side of this power cord.

On the left edge of the rear panel are a three-position switch and a six-pin DIN socket. This socket is the serial port through which the printer is connected to your computer. When one or more 1541 Disk Drives are in use, the printer should be daisychained with the computer system. Instead of plugging the printer directly into the computer, plug it into the serial port of the last disk drive.

The three-position switch at the left selects the unit's device number (it may be either 4 or 5) or activates a self-diagnostic test. During testing, the printer's entire character set is typed out repeatedly so that both the machine and the printout may be inspected for operational errors.

The VIC-1525 is fitted with a removable smoke gray plastic cover, which not only provides a dust barrier for protection of the printer's internal parts, but also offers some insulation from printing noise.

Looking into the printer with the cover removed, you'll see its working parts. Standing slightly higher than the rest of the printer body is the paper-feed assembly, a rod spanning the printer's width. On each end of the bar are the two adjustable sprocket units that hold the paper and feed it through.

Between the two sprocket/paper-holder assemblies are three paper supports, which can be spaced along the bar to keep the paper from jamming as it leaves the printer. To the far right of the paper-feed assembly, on the printer's main body, is a manual paper-feed dial, which rotates in only one direction (clockwise). A matching slot on the printer cover gives access to this dial while the cover is in place.

Closer to the front of the printer, and lower than the paper-feed assembly, are the print-head assembly and the ribbon platforms. The red LED (light-emitting diode) on the printer's name plate is the power-indicator light.

VIC-1526. In contrast to the 1525's angular design, the 1526 Dot-Matrix Printer is flat and boxy. Its power-supply cord and two six-pin DIN connectors are found on the rear panel; a rocker switch located on the printer's side panel turns power onto the unit.

A clear plastic removable cover on the top of the printer protects the printing assembly and paper. Behind the paper-feeding assembly rests a removable wire attachment that holds incoming fanfold paper. Unlike the 1525, the 1526 lets you advance paper manually, using a knob found on the printer's left side. When power is turned on, paper can also be advanced automatically by pressing the paper-feed button located on the upper right corner of the printer's front panel.

Like the 1525, the 1526 has adjustable sprocket units that adapt to various sizes of fanfold paper. However, rather than using a two-part ribbon cassette, the 1526 has a single-ribbon cartridge, which fits across the entire printing bay.

Connect your Commodore printer to your computer only after you have verified that both the printer and the computer are turned off. A six-pin DIN cable is provided with the printer for connecting the printer's serial port to the computer's serial port. Because both ends of this cable are identical, either can be plugged into the printer or computer. To make things simpler, the plugs are keyed and will only go in one way. Then, plug the power cord into a standard household outlet.

Ribbon Installation

VIC-1525. Before you use the 1525 Graphic Printer, both paper and a ribbon cassette must be installed. The ribbon cassette comes in two halves, which are installed on opposite sides of the print-head assembly's printing path. The ribbon is stretched between the cassette halves. Follow these steps to install the ribbon cassette.

1. Make sure that the printer is turned off.
2. Remove the printer's cover.
3. Remove the ribbon cassette from its sealed container.
4. Orient the ribbon cassette so that the two halves are side by side and the holding tab on the front edge of each half is facing you. There will be a curved lip on the outer edge of the bottom of each cassette half.
5. Gently pull the two cassette halves apart so that the ribbon is extended between them.
6. Slip the lipped rim of the left ribbon cassette half over the left edge of the printer's left cassette-holding platform (the flat silver platform with two holes in it).
7. Thread the far ribbon piece between the print head and the platen.
8. Thread the near ribbon piece between the swiveling ribbon cam and its backing piece.
9. Draw the right ribbon cassette half over to the right cassette-holding platform, being careful not to unthread the ribbon. Be sure there are no twists in the ribbon. Snap into place.
10. As with the right ribbon cassette half, slip the left cassette half's lipped rim over the platform and snap the cassette into place.

VIC-1526. Follow this guide to installing the 1526 Dot-Matrix Printer's ribbon cartridge.

1. Make sure that the printer is turned off.
2. Remove the printer's cover.
3. Remove the new ribbon cartridge from its container.
4. Orient the cartridge so that the small plastic knob is on top of the cartridge and the ribbon itself is closest to the printer.
5. Take up any slack in the ribbon by turning the plastic knob in the direction shown on the cartridge.
6. Fit the cartridge into the printing bay, beginning with the left edge of the cartridge. Tilt the cartridge until it is seated in place.
7. Lower the cartridge's right side until it is also seated.
8. Snap the cartridge firmly into place.
9. Again, turn the cartridge knob to take up slack in the ribbon and to bring the ribbon into place with the print head.

Paper Installation

VIC-1525. Installing the printer's tractor paper is a simple operation, and if the printer is left with the paper installed, you have to repeat the process only when the stack of fanfold paper is depleted. It is safer to turn the printer off during paper-installation operations.

A wide variety of paper is available in fanfold pin-feed style. "Clean perf" paper is ideal, because the perforations are so small that when the tractor-feed edges are removed, the remaining paper is virtually indistinguishable from standard typing paper. Some stationers now offer high-quality letterhead stationery in continuous pinfeed form.

The process for threading the paper into the printer is outlined below.

1. Remove the printer's cover.
2. Lift the first sheet of fanfold paper and bring it to the back of the printer.
3. Slide the paper along the sloping body of the printer and under the clear plastic paper guide.
4. Continue to push the paper through the printer until it appears between the platen and ribbon.

5. Approximate the width of the paper, and space the adjustable sprocket-unit assemblies accordingly.
6. Lift the paper holders up from their locked positions over the sprocket units.
7. Pull the paper up to the sprocket units and fit the paper's pin holes over the sprocket pins. Final sprocket-width adjustments can be made at this time.
8. Close the paper holders over the paper and sprocket units.
9. Advance the paper with the paper-feed dial until the paper hangs over the rear of the printer.
10. Replace the printer's cover.

VIC-1526. Insertion of tractor paper into the 1526 Dot-Matrix Printer is performed in a similar manner.

1. Remove the printer's cover.
2. Place the stack of fanfold paper either behind the printer, when the optional wire paper-holding attachment is not in place, or on top of this rack, when the attachment is in place.
3. Lift the first sheet of fanfold paper and bring it to the back of the printer.
4. Lift the paper holders up from their locked positions over the sprocket units.
5. Push the paper through the printer past the sprockets, around the platen, and up through the printer.
6. Align the incoming paper over the sprocket units and close the two paper holders over the sprockets and the paper edges.
7. Replace the printer's cover.

Power Up

VIC-1525. Once the ribbon and paper are installed, either printer can be turned on with the rest of the computing system. If you are using a 1525, be sure that the three-position switch on the printer's rear panel is set to the position marked 4. Then turn each part of the system on, being sure that the computer is turned on last.

First, the printer's power-indicator light will come on. Then the print-head assembly will travel to the center of the print path and then return to the left end of its path. If that doesn't happen, check the cable connections.

For an immediate demonstration and a sample of the printed character set, move the rear panel's three-position switch from the setting labeled 4 to the T position. This self-diagnostic test setting causes the printer to print its character set continuously until the three-position switch is moved from T to another setting (be sure the switch is returned to setting 4).

If the printed copy is too light or too dark, a print-head striking-strength adjustment can be made by repositioning the lever on the print-head assembly. Move the print head closer to the paper for darker print and farther away for lighter print.

VIC-1526. Turning on the 1526 is simple; there is no external device number setting for you to adjust.

To execute the 1526 print test, depress the paper-feed button while turning on the power. The printer's character set will be printed continuously until the power is again turned off. To restore normal operation, simply push the power switch by itself. Once again, be sure that both paper and the ribbon cartridge are installed before doing any printing.

Program Applications

The Commodore computer can give printing instructions to the printer only through an OPENed line of communication, usually from within a program. The OPEN command serves this purpose, accompanied by information that tells the computer how and where to send the signals. The OPEN command simply sets up a line of communication between the computer and its peripherals. Each peripheral can be given its own "phone number," in effect, and the OPEN command is like dialing the number. The generalized syntax for the OPEN command line is this:

OPEN file number, device number, secondary address

The file number chosen can be any number between 0 and 255, but note that a file number between 128 and 255 causes an extra linefeed after each printed line. The *same* file number must be used for any printing instructions you send later that call for a file number. In other words, to reach the same phone, use the same phone number. The file number is part of that phone number.

While the 1525's external device-number select switch permits a choice of device numbers, the 1526 is device number

4, unless it is changed internally. (Only a trained Commodore service technician should attempt this modification.) This allows you to use more than one printer at a time.

The secondary address specifies whether printing will be done in upper- or lowercase. A secondary address of 0 is the default value for this part of the OPEN command. This selects printing in uppercase and graphics. When a secondary address of 7 is chosen, printing is in upper- and lowercase. You might, for example, use OPEN 1,4,0 to OPEN a printer file for uppercase printing, with a file number of 1. It is acceptable to OPEN a printer file with 0 as the secondary address and then to select upper- and lowercase printing via program control.

LISTing a program on the printer is a simple matter of channeling normal screen output to the printer instead of to the TV screen or monitor. This is comparable to having your phone call transferred from one phone line to another. Once a file has been OPENed, the CMD command will perform this function. For example, CMD 9 sends screen output through an OPENed file numbered 9. With a program already in the computer's memory, type the word LIST and press the RETURN key. Rather than appearing on the screen, the program is LISTed on the printer, along with the usual READY prompt. All the computer's messages will be printed on the printer until the CMD command is canceled. The syntax for the CMD command is:

CMD file number

Just as the LIST command's actions are transferred to the printer after the CMD command is issued, the PRINT statement will also send its output to the printer.

Canceling the CMD command is a two-part operation. First, any remaining data is cleared out of the channel. Emptying the channel of extra data is accomplished with one of the most important printer commands, PRINT#. Then the file is CLOSEd. PRINT# is like saying, "That's all I have to say for now," and CLOSE is like hanging up the phone. These commands are followed only by the file number (for instance, PRINT# 12) when CLOSEing the file. After pressing RETURN, any data in file 12 will then be dumped onto the printer. Then, to finish CLOSEing file number 12, type CLOSE 12 and press RETURN.

Generally, this is the syntax for CLOSEing a file after using the CMD command:

PRINT# file number: CLOSE file number

The colon separating the two commands allows both instructions to be placed on a single line.

PRINT# has many more uses than just assisting in CLOSEing a file. This command can be followed (after the file number is stated) by data to be printed on the printer. Material to be printed can either be contained within quotation marks (just as if you were printing it on the screen with a PRINT statement) or text can be stored in a variable following the PRINT# command (for example, PRINT# file number, data). Note that a comma must follow the file number.

When presented in this form, text is printed on the printer in its standard form in either upper- or lowercase (depending on how the file was OPENed). However, Commodore's 1525 Graphic Printer is capable of many other printing formats. For the most part, the instructions to access these alternate print styles are found in the form of CHR$ codes, which accompany the PRINT# command.

More than one CHR$ code can be given with a single PRINT# command. The print controlling instructions follow this pattern:

PRINT# file number, CHR$(x) data

Table 4-1 shows each of the possible CHR$ printing control codes and the function it performs.

Table 4-1. CHR$ Printing Control Codes

Function	Code
Select double-width mode	CHR$(14)
Select standard-character mode	CHR$(15)
Select reverse-field printing	CHR$(18)
End reverse-field printing	CHR$(146)
Select upper/lower case mode	CHR$(17)
Select uppercase/graphics mode	CHR$(145)
Set custom-graphics mode	CHR$(8)
Repeat graphics	CHR$(26)
Tab setting	CHR$(16)
Set dot address for tab	CHR$(27)
Perform carriage return	CHR$(13)
Perform linefeed	CHR$(10)

Commodore's 1525 Graphic Printer makes use of both CHR$ codes and secondary addresses to control the output of text. But the 1526 Dot-Matrix Printer depends almost entirely on secondary addresses for formatting. For example, secondary address controls the number of lines to be printed on a page and the spacing between lines. Also, the printer can be made to take numeric or character-string data and print it in prescribed columns. Each of the secondary addresses is established by OPENing a file to the printer and sending the appropriate secondary-address code.

Table 4-2 shows the secondary-address codes of the 1526 Dot-Matrix Printer and the function each performs.

Table 4-2. Secondary-Address Codes For The 1526 Dot-Matrix Printer

Function	Secondary Address
Print data in uppercase/graphics mode	0
Print data as previously defined	1
Store formatted data	2
Set lines per page	3
Enable diagnostic messages	4
Define custom character	5
Set line spacing	6
Print data in upper/lower case	7
Defeat diagnostic messages	9
Reset printer	10

A PRINT# statement containing formatting data should follow any of these OPEN statements to transmit formatting information to the 1526 Dot-Matrix Printer.

Secondary address 2 lets you store formatting instructions in the printer. This will allow, for example, columns of numbers to be aligned on their decimal points.

It is even possible to send numeric data to the printer, where a fixed or floating dollar sign and decimal point will be added automatically. But special formatting characters must first be sent to the printer to establish a template for the incoming data. In the case of a template establishing a fixed dollar sign, a dollar sign character ($), plus space holders (represented by the 9 character) and a decimal point, are first sent to the printer through a channel OPENed with secondary address 2. It is best to perform this formatting action through numbered BASIC program lines, but the statements are discussed here individually.

The first statement is OPEN 1,4,2. The 1 is the file number and could be any number from 1 to 255. The printer's device number (4) is next, followed by the secondary address (2).

The actual formatting template is then sent with a PRINT# statement. The line PRINT# 1, "$999.99" prepares the printer to print the data (no matter how it receives it) with a dollar sign first, followed by three place holders, a decimal point, and two more place holders.

This formatting file (1) is then CLOSEd with CLOSE 1. The printer is ready to accept numeric data and it will print it in the established format.

Data can come from another file number (for example, 8) using the following commands:

```
OPEN 8, 4, 1
PRINT# 8, 56.29
CLOSE 8
```

The data, 56.29, does not have to be placed within quotation marks within the PRINT# statement. On the printer, the data should appear as $ 56.29. An empty space is found between the dollar sign and the first digit of the number. The formatting command holds open the hundreds column even though there is no digit there in the data. This makes for neater, more readable business printouts.

Changing Print Styles

Although changing the 1525 Graphic Printer's print style is as easy as adding a single control code to the PRINT# command, seeing these codes at work within an actual program makes it easier to understand them. This short BASIC program utilizes some of the most popular printing control codes. This program asks for the user's name and then prints the name in four different styles on the VIC-1525 Graphic Printer. Finally, a custom character is designed and printed underneath the name. It will run on either the VIC-20 or the Commodore 64.

```
1 OPEN 1,4
2 INPUT "YOUR FIRST NAME";A$
3 FOR X=1 TO 6:READ B
4 B$=B$+CHR$(B):NEXT X
5 PRINT#1,A$;"'S"
6 PRINT#1,CHR$(18) A$;"'S" CHR$(146)
7 PRINT#1,CHR$(14) A$;"'S" CHR$(15)
8 PRINT#1,CHR$(14) CHR$(18) A$;"'S" CHR$(15) CHR$(
  146)
9 FOR X=1 TO 80
10 PRINT#1, CHR$(8) B$;
11 NEXT X
12 PRINT#1, CHR$(15)
13 DATA 128,255,162,148,136,128
14 PRINT#1:CLOSE1
```

The program begins with an instruction to OPEN a channel to the printer (line 1). The number 1 has been chosen for the file number, followed by 4 as the 1525 Graphic Printer's device number. Line 2 prints a prompt on the screen for the entry of the user's first name to be stored in the variable A$ for later printing.

Lines 3 and 4 work together READing DATA for the creation of the custom character. Because there are six pieces of DATA (in line 13), the FOR loop in line 3 counts from 1 to 6. Within each execution of the loop, a single piece of DATA is READ (line 3). Each DATA element is added to the previously retrieved DATA to create the entire character in B$ in line 4. Line 4 also contains the loop's accompanying NEXT statement.

In line 5, the first printing is done on the printer. Since file number 1 has already been OPENed, the PRINT# command can now be used. The user's name, stored in A$, is printed normally, without CHR$ code alteration. A semicolon is placed after A$ so that the possessive apostrophe ('S) will follow the person's name. Note here that text to be printed is enclosed within quotation marks.

Then, in line 6, reverse-field printing is selected by CHR$(18). Notice that a comma, as usual, follows the PRINT#'s file number, but no special punctuation must surround the CHR$ code. The user's name is then printed. Instead of the black dots on a white field, as in the first PRINT# line, reverse field prints the name in white letters on a black field. To end this style of printing before the next line is encountered, CHR$(146) is printed.

Following the same syntax as line 6, line 7 causes the name to be printed in double-width mode with CHR$(14). Standard character size is resumed with the controlling code, CHR$(15). Line 8 combines both double-width and reverse-field printing by simply giving both codes before the text to be printed. Once again, both modes are turned off by their appropriate canceling codes.

Lines 9–11 perform the printing of the line of the custom graphics character. Because the printer has a maximum capability of an 80-character column width, line 9 begins a loop that will count from 1 to 80. In line 10, custom-graphics mode is selected, and the program uses the special DATA stored within the string variable, B$. A semicolon follows B$ so that subsequent printings of the character will be made side by side. Line 11 completes the loop started in line 9. Once again, CHR$(15) cancels either double-width or custom-graphics printing.

It is important to remember to reset the print style to normal once it has been altered. Otherwise, the printer will remember its previous code alteration. Line 13 holds DATA for the custom character and line 14 closes the file.

Custom Characters

It may be helpful here to briefly discuss the creation of custom characters and to see how their DATA is derived.

The DATA given in the preceding example creates a right-pointing triangle. In order to conform to the standard character size, the triangle shape was kept within a 6 × 7 dot matrix.

Each X in the grid in Figure 4-1 represents a dot that will be created by the print head. Because the 1525's print head consists of seven vertically arranged pins, each column of the grid represents one strike of the print head. The numbers of the DATA statement (line 13) each represent a column of dots.

Figuring the value for a custom character is done one column at a time. Each row of the grid has its own place value. For every X on the grid, add in the place value for that row, until the entire column has been calculated. Finally, add 128 to this number. The column data is placed in a DATA statement in the grid order, from left to right. Theoretically, a custom character can be as wide as the printer will allow (80-character columns × 6 dots each = 480 dots wide.) A separate piece of data would represent each column.

Figure 4-1. A VIC-1525 Custom Character

```
. X . . . .        1
. X X . . .        2
. X . X . .        4
. X . . X .        8
. X . X . .       16
. X X . . .       32
. X . . . .       64
                +128
```

The 1526 Dot-Matrix Printer is also capable of creating custom characters. However, a difference in the process for creating these characters makes the program listing for the Graphic Printer incompatible with the Dot-Matrix Printer. Most notably, the 8 × 8 dot matrix of the 1526 calls for the design of a different character pattern.

The calculation of the character data is performed exactly as it was for characters for the 1525 Graphic Printer, except that each of the 1526 Dot-Matrix Printer's character rows is assigned a new value, as shown in Figure 4-2. The data for this particular character would be 0, 0, 254, 68, 40, 16, 0, 0.

Figure 4-2. A VIC-1526 Custom Character

```
. . X . . . . .       128
. . X X . . . .        64
. . X . X . . .        32
. . X . . X . .        16
. . X . X . . .         8
. . X X . . . .         4
. . X . . . . .         2
. . . . . . . .         1
```

Because the 1526 Dot-Matrix Printer uses secondary addresses for most print-formatting operations, the same holds true for the creation of custom characters. A separate channel is OPENed to the printer to select character creation and to store the character's value. The following sample program lines give an example of the creation of a custom character using the DATA provided above.

```
10 OPEN 9,4,5
```

This line opens a character-creating channel to the printer (secondary address 5).

```
20 FOR X=1 TO 8:READ B
30 B$=B$+CHR$(B):NEXT X
40 DATA 0,0,254,68,40,16,0,0
50 PRINT#9,B$
```

These lines collect the DATA into string variable B$ and send this character information to the printer.

```
60 OPEN 7,4
70 PRINT#7,CHR$(254)
80 CLOSE 9:CLOSE 7
```

This OPENs a new channel to the printer for actual printing of the character, prints the character by the instruction code CHR$(254), and CLOSEs both printer channels.

Reference Section

Syntax of Printer Functions

Function	Command
Close a file	CLOSE file number
Open a file	OPEN file number, device number, secondary address
Send data	PRINT# file number, data*
Transfer output to printer	CMD file number*

* An OPEN command is required prior to these functions and a CLOSE afterward. *Do not include the asterisk (*) as part of the command.*

Glossary

Bidirectional. The ability of a printer to print while the print head is traveling in either direction.

Correspondence quality. The print produced by a dot-matrix printer that is considered to be of adequate quality for professional correspondence.

Daisywheel. The disk-shaped printing element of the daisywheel printer which carries the character types.

Dot-matrix print head. The set of pins that strike the ribbon of a dot-matrix printer to create the image of a character.

Draft quality. The print produced by a dot-matrix printer if it is not adequate for professional correspondence.

Fanfold paper. Pin-feed computer-printer paper which is folded along perforations for easy, compact storage; also called *tractor paper.*

Friction feed. The method of pulling paper through a printer by means of friction rather than with tractor pins and specialized paper; also called *pressure feed.*

Hard copy. A printout.

Letter quality. Printed text which is indistinguishable from that produced by a typewriter.

Tractor feed. The method of feeding specialized paper through a printer that uses a pin-feed mechanism.

Unidirectional. A method of printing in which characters can be printed only when the print head moves left to right.

Chapter 5

The VIC 1520 Printer/Plotter

Chapter 5

The VIC-1520 Printer/Plotter

The standard dot-matrix printer is adequate for most text and graphics printing. However, for some graphics applications, you may need graphs and charts with a cleaner appearance. Dot-matrix graphics cannot easily and accurately plot specific x and y coordinates, for instance, and they cannot produce truly continuous lines. No matter how closely the dots of a dot-matrix printer are placed, they are still dots—and if a smooth diagonal line is desired, the dot-matrix printer will not do the job. For these tasks, you need something that can plot continuous lines or individual points at *any* point on a sheet of paper—a printer/plotter.

The print head of a printer/plotter is not made up of wires (like that of a dot-matrix printer) or of type elements (like that of a daisywheel printer). Instead, it uses pens. Usually, they are just like regular ink pens but are more precise. A pen holder moves the pen back and forth along a moving arm, raising and lowering the pen to draw the figures on the page.

The VIC-1520 is a drum-type printer/plotter. It rolls the paper back and forth on a platen while the pen holder moves side to side. This means you can move the pen almost anywhere you want. Although the pen holder is restricted in its movement by the width of the paper, the continual up and down rolling of the paper provides a canvas with virtually unlimited y coordinates (up-and-down movement). The graphics produced by a printer/plotter are far superior to those of other available printing methods, and the alphanumeric characters are more readable than those of many dot-matrix printers as well.

What Can the 1520 Printer/Plotter Do?

High-resolution graphics is the 1520's specialty. Again, the plotting is performed by the *combined* movement of the pen holder and the paper following the standard plotter x,y coordinate system: the pen holder position is on the x-axis and the paper movement is on the y-axis.

The ball-point pens produce a line 0.2mm wide. Even though this resolution is remarkable, perhaps the best feature of the pens is that each of them is a different color (blue, green, red, and black), and each of the colors can be used individually through program control. This gives you full color graphics with a minimum of effort.

Of course, the 1520 can also print alphanumeric characters (upper- and lowercase). Four character sizes are available, with text in any of the four colors. In fact, under program control, the 1520 can combine color, graphics, and text on the same line, a feature that is essential in the production of business charts.

While these capabilities might sound attractive, the real test is in direct applications. Since Commodore made the 1520 Printer/Plotter a "smart" peripheral, it is easier to use and you can control it with relatively simple programming.

A Tour of the 1520 Printer/Plotter

Since most of the 1520's features are controlled through software, it has little need for manual controls. The few controls it has are on the top, the right side panel, and the rear panel.

Along the front of the top side are three switches: paper feed, color change, and pen change (left to right). Of these three, color change is the only one that can be performed in a program; you can also physically rotate the pen holder to put a different color into printing position.

The paper feed is used to move paper manually through the plotter. This might be applicable when you need extra space between graphics images. Finally, the pen change switch prepares the plotter for replacement of the pens. Next to these switches is the red power-indicator light. Completing the top side is the printer mechanism cover. This lid not only protects the delicate pen holder, but is also designed to make tearing out the printed sheet easier. It serves too as a serrated tearing point for paper removal.

The rear panel has four features. From left to right, they are the fuse holder, the power-cord connector, the paper-holder assembly, and the Commodore serial port. The paper-holder assembly is protected by a hinged, smoked-plastic dust cover. When printing, be sure that the cover is rotated out of the paper's way.

The power switch is the only control located on the right side panel.

If you remove the printer mechanism cover (by carefully pulling up on its rear edge), there are several internal points of interest. These include the platen, the pen holder, and the pen-change lever. The platen is a black rubber roller that acts as a surface for the pens to write against. Two brass wheels, located on either side of the platen, guide the paper through the printer mechanism. The pen holder is the white plastic carriage that moves laterally across the printer mechanism.

To insert or remove one of the four pens, use the pen-change lever. This is the black rod positioned on the right-hand side of the printer mechanism.

Using the 1520 Printer/Plotter

Remember, when selecting a site for the plotter, to find a perfectly level spot. This is necessary because of the way in which a drum-type plotter works. Fortunately, the relatively small 1520 Printer/Plotter does not take up much room.

Two cables are included with the plotter—a black serial cord and a gray power cord. The gray power cord is plugged into the power-cord connector on the rear panel. All power should be off (both the plotter's and the computer's) when this connection is attempted. The flat, three-slotted end of the power cord is the connector for the plotter, while the familiar three-pronged end plugs into a regular household outlet.

Next, with power still off, connect the serial cable to both the plotter and the computer. All the Commodore serial ports are notched to prevent any incorrect connections. Either end of the serial cable can be plugged into either Commodore serial port. Incidentally, the 1520 Printer/Plotter can also be used with a Commodore computer that is equipped with a 1541 Disk Drive. Instead of plugging the serial cable into the computer's serial port, plug it into the unused serial port on the rear panel of the disk drive. This daisychaining can be used with up to four disk drives and one plotter.

Occasionally, of course, you'll need to change the paper and the pens in the 1520. To change the paper, use this procedure:

1. Turn off the plotter and the computer.
2. Remove the printer mechanism cover.
3. Cut a straight edge along the paper's end.
4. Slide the 4-1/2-inch-wide paper onto the paper-holder spool.

5. Place the spool on the support arms of the paper-holder assembly. Make sure that the paper unwinds from its bottom.
6. Feed the paper's end into the rear of the print mechanism.
7. Manually rotate the platen until the paper is visible.
8. Make sure that the paper is properly fed between the platen and the brass guide wheels.
9. Replace the print mechanism cover.
10. Turn on the plotter, then turn on the computer.

You can save some money by purchasing paper in bulk rolls from business-supply stores. The paper's width must be 4-1/2 inches, but any length is fine. It's a simple matter to remove several feet of paper as needed. This method also eliminates much of the drag that results when the printer/plotter must pull against a heavy roll of paper.

To replace pens, follow this procedure:

1. Turn on the plotter, then turn on the computer.
2. Remove the print-mechanism cover.
3. Push the color-change switch until the proper pen color is on top of the pen holder.
4. Push the pen-change switch.
5. Press on the white plastic portion of the pen-change lever.
6. Remove the pen.
7. Remove the colored plastic cap from the new pen.
8. Insert the new pen, with the ball point entering the pen holder first.
9. Push down on the pen's barrel until it clicks into place.
10. Repeat steps 3–9 until all pens have been changed.

This same procedure should be used for initially installing pens into the 1520 Printer/Plotter. Colored tabs indicate which pens should be inserted into each of the pen holder's slots.

A special test is triggered when either the plotter or the computer is turned on. This test consists of printing four different-colored boxes along the right-hand side of the paper. The boxes should be blue, green, red, and black, from left to right. The test checks several key plotter features. First, it checks to see if the pens are properly installed. If the colors are not in the correct order, the pens have been inserted in the wrong position. Second, it checks to see if the power cable was properly connected (the test won't happen at all if the cable is not in-

Finally, if the plotter is attached to the computer, the test will indicate whether the serial connection(s) are properly made.

Program Applications

To operate the 1520 Printer/Plotter under program control, a special communications channel must be established. This channel serves as a link between the computer and the plotter. Here is a generalized syntax for opening this channel:

OPEN file number, device number, secondary address

As an example, the line OPEN 9,6,1 would make a communications link (OPEN 9) with the plotter (6) and instruct the plotter to prepare for x,y plotting (1).

With the communications link open, data can be sent to the plotter. Two commands are added for data transmission: CMD and PRINT#. Only PRINT# sends data in the typical manner (similar to that of a PRINT statement); CMD transfers output from the monitor or television screen to the plotter. A generalized syntax for each of these commands follows:

CMD file number
PRINT# file number, (data)

The most common use for CMD is in LISTing a program on the plotter. Using the already OPENed file above (9), the syntax for LISTing a program would be CMD 9:LIST.

Data sent with PRINT# allows access to all the plotter's functions. In the generalized syntax of the PRINT# command, this data is supplemental information and designated by a DATA statement. This data can range from none (e.g., plotter reset) to a subcommand and coordinates (e.g., plot x,y data). Available functions are print characters, plot x,y data, change pen color, select character size, rotate characters, go to broken-line mode, select upper/lowercase characters, and reset plotter.

Each of these functions is selected or activated through the secondary address of the OPEN command. However, the various choices (e.g., pen color) and the actual data (e.g., x,y coordinates) are controlled through PRINT#. For example, the syntax for printing characters on an already OPENed file (9) and properly identified secondary address would be PRINT# 9, "MY NAME".

Finally, at the end of data transmission the communications link must be broken. The CLOSE command ends all communication with a specified file. This is a generalized syntax:

CLOSE file number

A special condition exists if the CMD command has been used with an OPEN file. The CLOSE command must be preceded by a PRINT# command. This is necessary for restoring the output to the monitor or TV screen. Therefore, in the previous program LISTing example, the syntax for CLOSEing the file would be PRINT# 9: CLOSE 9.

The use of the colon between PRINT# and CLOSE allows multistatement lines. Of course, each of these commands could have been entered separately on individual lines with the same results.

The microprocessor inside the 1520 Printer/Plotter has its own minilanguage, which enables the plotter to perform complex functions with a minimum of programming. Of the previously described commands, only OPEN and PRINT# are needed to control all of the plotter's functions. The secondary address of the OPEN command and the supplemental data portion of the PRINT# command work together to control the 1520. Tables 5-1 and 5-2 list all of the 1520 Printer/Plotter's functions and their accompanying controls.

In a typical program, the OPEN command will activate the plotter function, and the PRINT# command will manipulate the parameters of the chosen function. Table 5-3 lists all of the 1520 Printer/Plotter's function parameters and their associated control numbers.

Table 5-1. Functions and OPEN Secondary Addresses

Function	OPEN secondary address
Print characters	0
Plot x,y data	1
Change pen color	2
Select character size	3
Rotate characters	4
Broken-line mode	5
Select upper/lowercase characters	6
Plotter reset	7

Table 5-2. Functions and PRINT# Secondary Addresses

Function	PRINT# supplemental data
Print characters	,data
Plot x,y data	,"subcommand", x coordinate, y coordinate
Change pen color	,color number
Select character size	,character size number
Rotate characters	,character rotation number
Broken-line mode	,broken line number
Select upper/lowercase characters	,upper/lowercase number
Plotter reset	no data

Table 5-3. Function Parameters and Associated Control Numbers

Function	Parameter	Number
Print characters	None	None
Plot x,y data	Move to start point	Subcommand H*
	Set relative origin	Subcommand I*
	Move to absolute origin (pen up)	Subcommand M*
	Draw to point from absolute origin	Subcommand D*
	Move to point from relative origin (pen up)	Subcommand R*
	Draw to point from relative origin	Subcommand J*
Change pen color	Black	0
	Blue	1
	Green	2
	Red	3
Select character size	80 characters/line	0
	40 characters/line	1
	20 characters/line	2
	10 characters/line	3
Rotate characters	Normal (nonrotated)	0
	90⁰ rotation to the right	1
Broken-line mode	Solid line	0
	Smallest line divisions	1
	Largest line divisions†	15
Select characters upper/lowercase	Shifted lowercase	0
	Shifted uppercase	1
Plotter reset	None	None

* The subcommand is followed by the x and y coordinates, if needed. Subcommands "H" and "I" do not require coordinates.

† There are 15 line-division values available in all. Each one is given a whole number from 1 to 15.

Do not include the asterisks (*) as part of the command.

Even though these parameters are fairly easy to understand, many of their effects are easier to visualize in the context of a program. The following 20-line program will create randomly colored, randomly designed, individually labeled geometric flowers when using either the VIC or the 64. One use of this program might be to make personalized nametags for school children.

```
1 OPEN 1,6,Ø
2 OPEN 2,6,1
3 OPEN 3,6,2
4 OPEN 4,6,3:PRINT# 4,3
5 OPEN 5,6,7
6 INPUT "YOUR FIRST NAME";A$
7 PRINT#2,"M",24Ø,Ø:PRINT#2,"I"
8 P=INT(RND(Ø)*3)
9 PRINT# 3,P
1Ø FOR A=1 TO 5Ø
11 X=1+RND(1ØØ):Y=1+RND(1ØØ)
12 X=24Ø+X*COS(2/X*A*[PI]):Y=1ØØ+Y*SIN(2/Y*A*[PI])
13 W=1ØØ+X*SIN(2/X*A*[PI]):Z=1ØØ+Y*SIN(2/Y*A*[PI])
14 PRINT#2,"D",X,Y:PRINT#2,"M",W,Z
15 NEXT A
16 PRINT#1:PRINT#1:PRINT#1
17 PRINT#1,A$;"'S"
18 PRINT#5
19 CLOSE1:CLOSE2:CLOSE3:CLOSE4
2Ø CLOSE5
```

(Note lines 12 and 13 in the program above; to enter [PI], hold down the SHIFT key and press ↑. It will appear on the screen as a PI sign.)

When programming with the 1520 Printer/Plotter, it is a good idea to include every secondary address or parameter, even when the value is zero. Although a zero value can be left out, this practice will help you avoid mistakes when turning on and off functions. In the long run this will help you avoid confusion in the program-debugging stages. As you become more skilled in your programming, though, this will become a matter of taste, since the zeros are not necessary for the printer/plotter.

This is the case in line 1; the secondary address for printing characters is 0. Of the remaining OPEN commands, line 2 creates a channel for plotting x,y data, line 3 changes pen color, line 4 selects character size and chooses ten characters per line (,3), and line 5 is a channel for resetting the plotter.

After line 6 receives the user's first name, line 7 moves the pen to coordinates 240,0 ("M") and sets the relative origin to this point ("I"). A random number from 0 to 3 is generated in line 8 and used for changing the pen color in line 9. The FOR loop beginning in line 10 makes the operation of lines 11–14 repeat 50 times.

Once again, random numbers are produced (line 11), and the results are used in lines 12 and 13. The four numbers from lines 12 and 13 are used as plotting coordinates in line 14. First, the x,y coordinates are used for drawing a line ("D"); then the w,z coordinates are used for moving ("M") to a new location. Line 15 serves as the end of the loop started in line 10.

Line 16 prints three blank lines between the flower and the user's name. Line 17 takes the name inputted in line 6 and prints it with the addition of an apostrophe and letter *S* ('S). Now, it's the plotter's turn to "sign" the flower picture. This is done with the same four-box test that is performed when you turn on the plotter. Line 18 resets the plotter, which causes the test to be printed. Finally, lines 19 and 20 close the communications channels that were established in lines 1–5.

Many variations are possible, and only slight modifications of the program are necessary to make changes. An interesting three-dimensional design can be created by altering the upper limit of the FOR/NEXT loop in line 10. Try changing the 50 to 100 (10 FOR A = 1 TO 100) or to 150 (10 FOR A = 1 TO 150) and note the different results.

Incidentally, one of the best ways to enjoy all the capabilities of the 1520 Printer/Plotter is to use the computer language Logo. Commodore Logo was specially created with the 1520 in mind. A program included with Commodore Logo makes control of the plotter as easy as single-word commands. That makes the 1520 as flexible as the Logo turtle, but the paper printout makes the language much more tangible.

Reference Section

Syntax of 1520 Printer/Plotter Functions

A generalized syntax is used for each function. An OPEN command should be given before each, and all should be followed by CLOSE.

Function	Command
Close a file	CLOSE file number
Open a file	OPEN file number, device number, secondary address
Send data	PRINT# file number, (data)

Special Secondary-Address Data Controls

Function	Secondary Address of OPEN	PRINT# format
Print characters	0	PRINT# file number, data
Plot x,y data	1	PRINT# file number, "subcommand",x,y
Change pen color	2	PRINT# file number, pen color number
Select character size	3	PRINT# file number, character size number
Rotate characters	4	PRINT# file number, character rotation number
Broken-line mode	5	PRINT# file number, broken line number
Select upper/lowercase characters	6	PRINT# file number, upper/lowercase number
Plotter reset	7	PRINT# file number, no data

Special Secondary-Address Parameter-Control Subcommands For Plot X,Y Data Functions

Parameter	Value
Move to start plot point	PRINT# file number "H",x,y
Set relative plot origin	PRINT# file number "I",x,y
Move to absolute origin	PRINT# file number "M",x,y
Draw to point from absolute origin	PRINT# file number "D",x,y
Move to point from relative origin	PRINT# file number "R",x,y
Draw to point from relative origin	PRINT# file number "J",x,y

Glossary

Commodore serial. A nonstandard serial port. Serial ports transmit information a bit at a time.

Daisywheel printer. A letter-quality printer that uses regular type for producing characters.

Dot-matrix printer. A printer that uses a series of small wires arranged in a line to produce characters.

Drum plotter. A plotter where both the paper and the pen move in unison to plot x,y coordinates.

Flatbed plotter. A plotter where the pen is attached to a moving arm, but the paper remains in a fixed position to plot x,y coordinates.

Plot. To draw data points on a page with reference to horizontal and vertical coordinates.

X,Y coordinates. A pair of numbers used to designate the horizontal (x) and the vertical (y) placement of a given point.

Chapter 6

The VIC-1011A RS-232C Interface

Chapter 6

The VIC-1011A RS-232C Interface

Standardization is hard to find in the home computer industry, but one of the most successful attempts has been with interfacing. There are several common interfacing schemes, but by far the most popular is the RS-232C serial interface (not to be confused with the Commodore serial interface).

One feature of the RS-232C interface is that it can *handshake* with a variety of devices. Because the serial interface was prepared originally for modem use, one computer was assumed to be the terminal. But most peripherals (such as printers) don't know how to act like a terminal, so they'll tell the computer to stop sending data when their buffer is filled. This sounds primitive and slow, but an RS-232C interface can alter the computer baud rate (the speed at which data is transferred); this results in data transmission that is several times faster than with a parallel interface.

As a result, the RS-232C interface can connect what is called the host computer to modems, printers, and other computers. Particularly in the areas of printer interfacing and computer-to-computer communication, an RS-232C interface can be extremely useful—and with the VIC-1011A RS-232C Interface both the VIC and the 64 can use this standard serial interface.

What Can the VIC-1011A RS-232C Interface Do?

Under some circumstances, the Commodore peripheral line might be unable to satisfy your demands. For instance, what if you want a faster print speed or need letter-quality printing? The VIC-1011A RS-232C Interface gives the Commodore computer an industry-standard port, to which other manufacturers' peripherals can be connected.

Another area of increasing interest to the average computer owner is *downloading*. This is the process of transferring data between two computers. Using the 1011A Interface Cartridge, a link can be established between a companion computer and a Commodore computer. Then any data held in one computer can be placed in the other for processing, storage, and so on.

The beauty of all of this is that the RS-232C port can be added or removed whenever it is needed. This makes using the port on the Commodore computers easier; it also makes it easy to make a nonstandard Commodore computer compatible with products from other companies.

Using the VIC-1011A RS-232C Interface

With no moving parts, lights, or switches, the 1011A is simple to install. But to use it efficiently, you'll need to understand the pin assignments of the RS-232C connector.

These pins are accessed through a special plug—an industry-standard DB-25—and a length of ribbon cable. You will need two plugs, one for each end. For greatest flexibility, the plugs should be easy to dismantle, so you will want to try various pin combinations. Next, connect the two DB-25 plugs with a suitable length of flat, 25-conductor ribbon cable. Even though as much as 50 feet of ribbon cable *could* be used, try to use a more modest length.

Determining pin assignments is simplified on the 1011A since each pin number is written prominently on the face of the RS-232C connector. Although there are 25 pins, only 9 are actual working pins. The rest have no connection and therefore no function. The 9 active pins are identified in Table 6-1.

Table 6-1. VIC-1011A RS-232C Interface Pins

Pin	Assignment
1	Protective ground
2	Transmit data, TXD
3	Receive data, RSD
4	Request to send, RTS
5	Clear to send, CTS
6	Data set ready, DSR
7	Logical ground
8	Data carrier detect, DCD
20	Data terminal ready, DTR

Using the interface is simply a matter of connecting to the proper pins. Which pins to use will vary for different applications and different peripherals. But in any case, note from these pin assignments that pin 2 would be used for sending data (to a printer, for instance) and pin 3, for receiving data (from another computer).

After the proper cabling has been constructed, you can install the 1011A Interface. It plugs into the user port. Refer to chapter 1 if you want to know more about the user port.

The actual steps for connection are straightforward:

1. Turn off the computer and all peripherals.
2. Insert the 1011A RS-232C Interface Cartridge into the Commodore user port.
3. Plug one DB-25 plug (with its predetermined cable configuration) into the RS-232C connector on the 1011A Interface.
4. Plug the other DB-25 plug into the serial port of the peripheral.
5. Following the proper starting procedures (i.e., computer last), turn on the system.

Take care when you insert and remove the 1011A Interface. The user port connector is actually part of the computer's circuit board and rough handling can seriously damage the computer. Gentle, even pressure will overcome any stubborn resistance. Also, inspect the contacts (both on the user port and on the 1011A Interface) for metallic deposits. These can degrade the signal and make the RS-232C port appear to be malfunctioning.

As long as the user port is not needed, leave the 1011A Interface in place. Even if the RS-232C interface is not being used, it can still remain attached to the computer. This will prevent unnecessary wear and tear.

Program Applications for the VIC-1011A RS-232C Interface

Six parameters are controlled by the 1011A Interface: baud rate, stop bit, word length, handshake, parity, and duplex. Each of these factors is adjustable, but only baud rate is frequently altered.

Baud rate is the speed of data transmission, measured in bits per second (bps). With the 1011A Interface, baud rates of 50–2400 bps can be selected. These speed rates must be matched to the peripheral's ability to accept (or transmit) data.

As its name implies, *stop bit* is used for signaling the end of a character. It is in the form of a transmitted blank bit that follows each character's bit data burst. For example, if a character consists of eight bits of data, the addition of one stop bit would give the total of nine bits of data. A choice of either one or two stop bits is available.

Word length may be the most confusing of these parameters, perhaps because it is character length that is determined, not word length. The most common character lengths are seven and eight bits, but the RS-232C port can handle from five- to eight-bit characters.

Handshaking is the computer's way of telling peripherals to wait. Similarly, if the computer is sending out data faster than a peripheral can handle it, the peripheral will tell the computer to wait. This handshaking between computer and peripheral reduces the possibility of data loss. With the 1011A Interface, handshaking is either on or off. Most often it will be on, often referred to as XON.

To insure verification of transmitted data, *parity* allows for special bits that are checked at the data destination. This error checking can be defined as even, odd, mark, or space parity. In each case the number of on bits is inspected and verified against the expected value. Parity can be disabled on the 1011A Interface.

Duplex is another method for checking the data that is being transmitted. By using duplex the computer's message can be reflected back to the computer. This mirroring is useful in some modem applications, but it is of little use in most others. The 1011A Interface permits full or half duplex selection.

All six of these parameters are set through two registers. The *control* register sets the baud rate, stop bit, and word length, while the *command* register controls handshaking, parity, and duplex. The values for these registers are determined by selective bit programming, turning on and off the individual bits in each of the two registers. Once the desired conditions have been established, they are fixed until the computer is turned off.

Both the control register and the command register consist of a single, one-byte character. Each bit (there are eight) of the register byte is turned on or off to control individual parameters. When programming on a Commodore computer in BASIC, this final register bit pattern is defined with decimal numbers from 0 to 255.

In the control register the bit patterns shown in Table 6-2 specify various baud rates, stop bits, and word lengths.

The final number in the control register is the sum of the desired parameters' register values. For example, for a baud rate of 1200, one stop bit, and a seven-bit word length, you would use a control register value of 40. In most instances, only the baud rate will have to be set, since the usual stop bit and word length values are zero (one stop bit and an eight-bit word length).

In a similar fashion, the command register governs the handshake, parity, and duplex. These values are summarized in Table 6-3.

As with the control register, the final command register value is a sum of all of the desired parameters' register values. However, with the exception of handshaking, most of the command register options will have the value zero. In other words, no parity but full duplex is a frequently used arrangement. If handshaking were also deemed unnecessary, the command register would have a final value of zero. For this reason, the inclusion of the command register is not mandatory.

Table 6-2. Control Register Values for Baud Rate, Stop Bit, and Word Length

Control Register	
Baud Rate	**Register Value**
50	1
75	2
110	3
134.5	4
150	5
300	6
600	7
1200	8
1800	9
2400	10
Stop Bit	**Register Value**
1 stop bit	0
2 stop bits	128
Word Length	**Register Value**
5 bit	96
6 bit	64
7 bit	32
8 bit	0

Table 6-3. Command Register Values for Handshaking, Parity, and Duplex

Command Register	
Handshake	**Register Value**
XOFF	0
XON	1
Parity	**Register Value**
None	0
Odd	32
Even	96
Mark	160
Space	224
Duplex	**Register Value**
Full	0
Half	16

Operating the RS-232C Port

To operate the RS-232C port, these two registers are applied to a special communications channel. This channel is used for establishing a link between the computer and the peripheral. A generalized syntax for opening this channel would be as follows:

OPEN file number, device number, control register command register

In the actual application, the following line would make a communications link (OPEN 1) with the RS-232C port (2), set the baud rate to 300, use a single stop bit, have a seven-bit word length (CHR$(38)), set handshaking, disable parity, and use full duplex (CHR$(1)):

OPEN 1,2,CHR$(38) CHR$(1)

As this example illustrates, the control and command registers are defined with character codes. Each character code represents one of the registers. The character code that is used is one corresponding to the sum of the register values. If handshaking was not required, the command register character code could have been removed and the line would look like this:

OPEN 1,2,CHR$(38)

The Commodore computer requires 512 bytes for OPENing a channel. A CLR is performed to obtain this space, and it's important to keep this in mind when designing programs. Conflict might arise between the DIM statement (and other variables) and the OPEN command. If DIM is performed first, the OPEN would steal memory from it. Therefore, it is best to always OPEN a channel prior to using DIM or making variable assignments.

Once the communications link is ready, data can be either transmitted or received. Three general commands are used with these two functions. However, the use of these commands will vary depending on the peripheral.

After the communications link has been OPENed, data can be transmitted with either CMD or PRINT#. Only PRINT# transmits data in a conventional sense. The CMD command transfers output from the monitor or television screen to the peripheral. A generalized syntax for each of these commands follows:

CMD file number
PRINT# file number, data

A common application of CMD is to LIST programs to an RS-232C printer instead of to the screen. Using an already OPENed file (4), the syntax would be CMD 4:LIST (followed by pressing RETURN).

On the other hand, PRINT# acts like a regular PRINT statement, except that all of the data is directed to the peripheral. This data can be in as many different formats as with a PRINT statement. The syntax for the most common format on an already OPENed file (4) would be PRINT# 4, "HI THERE" (followed by RETURN).

To receive data, only one command is necessary (GET#). In function, GET# is similar to its GET statement counterpart. Here is a generalized syntax for using GET#:

GET# file number, string variable

Many different programming applications can be found for GET#. Since only a single character at a time is received by GET#, some sort of loop will have to be used. In the following example, an already OPENed channel (5) is used for receiving data. Error checking and other conditionals have been removed for simplicity.

```
10 GET# 5,A$
11 PRINT A$;
12 GOTO 10
```

Finally, at the conclusion of all data transmissions and receptions, the communications link must be broken. The CLOSE command terminates all communication with a specified file. A generalized syntax is:

CLOSE file number

One special condition does exist, however. Whenever the CMD command is used, it must be preceded with a PRINT# command. This is necessary to restore the output to the screen. Therefore, after LISTing a program with the CMD command (with file number 5), the syntax for ending transmission would be PRINT# 5:CLOSE 5.

The use of the colon between the PRINT# and CLOSE commands allows more than one statement to be executed from the same line on the screen. Both of these commands could have been entered individually (followed by RETURN) with the same results.

At the conclusion of a transmission, some data might remain in the computer's internal buffer. This should be cleared out prior to a CLOSE command. In practice, a program could be designed to look for a special status indicator that would indicate the end of data. Alternatively, a series of blank PRINT# commands could be issued to clear out the buffer.

But what about practical applications for the RS-232C port? Of the many possible uses, connection to an RS-232C printer and connection to another computer are the most interesting and the most common. All the syntax and register values remain the same—the greatest difficulty is with the hardware. Since all RS-232C ports are identical (theoretically), the interconnecting ribbon cable can't generally run from pin X to pin X. Some minor rewiring is usually necessary. However, only the cable connections need to be changed. You will not need to modify the RS-232C port.

Due to minor variations in RS-232C ports found on various printers and computers, no fixed wiring scheme can be given. Instead, you'll have to study a pin assignment chart for each particular peripheral. Then, based on the peripheral's expected duties (will it be transmitting or receiving), the appropriate ribbon cable connections can be made.

Rewiring must be done with caution. If a wrong connection has been made, harm might come to the peripheral or the computer. For this reason we recommend that you have a qualified technician make up the necessary cables.

Reference Section

Syntax of VIC-1011A Functions

A generalized syntax is used for each function.

Function	Command
Close a file	CLOSE file number
Close a file after a CMD	PRINT# file number: CLOSE file number
List a program	CMD file number:LIST*
Open a file	OPEN file number, device number, control register, command register
Receive data	GET# file number, string variable*
Transfer output to the peripheral	CMD file number*
Transmit data	PRINT# file number, data*

* An OPEN command is required prior to these functions, and a CLOSE afterward. *Do not include the asterisk (*) as part of the command.*

Glossary

Baud rate. Speed of data transmission measured in bits per second (bps).

Centronics. A parallel-type interface.

DB-25. An RS-232C 25-pin connector.

Downloading. The transfer of data from one computer to another.

Duplex. An error-checking feature for data transmission. The data is repeated back to the sender. Can be either full or half duplex.

Halfplex. Another name for half duplex.

Handshake. Used by a peripheral to temporarily halt the data flow. Can be on or off.

IEEE. An *I*nstitute of *E*lectrical and *E*lectronics *E*ngineers parallel-type interface.

Parallel. Data is transmitted or received a byte (eight bits) at a time over eight separate lines.

Parity. An error-checking feature for data transmission. Selective bits are manipulated and then examined by the receiver. Can be even, odd, mark, or space.

RS-232C. A serial-type interface.

Serial. Data is transmitted or received a bit at a time over one single line.

Stop bit. Signals the end of a character. Can be one or two stop bits.

Word length. The number of bits constituting a given character. Can be five, six, seven, or eight bits.

Chapter 7

The VIC-1111 16K Memory Expander for the VIC-20

Chapter 7

The VIC-1111 16K Memory Expander for the VIC-20

There are two kinds of computer memory. ROM (read only memory) stores data or instructions in such a way that they can be read only by the computer. But a personal computer would be of little use without some way for you to store instructions, and that's the function of RAM (random access memory). RAM holds new instructions written to it by the user and allows those instructions to be read by the computer. RAM is a *volatile* type of memory—its contents are lost as soon as the computer's power is turned off.

The amount of room available in RAM affects the length and complexity that a program can have. The larger the RAM area, the more programming instructions can be stored there.

What Can the VIC-1111 Do?

Although the VIC-20 is billed as a computer with 5K of memory (this value represents RAM), certain functions of the computer's ROM require some RAM for their own uses. This leaves 3.5K of RAM for programs when the VIC is in its normal state.

With all the features built into the VIC-20, this 3.5K memory limitation can impose a significant limit on a programmer. In fact, it may often seem that just as a program begins to take shape, you must drastically trim down the project due to memory restrictions. This limited memory requires you to hone your programming techniques. But if you feel you have learned how to get the most from every byte of memory in the computer and you still need more, it may be time to upgrade your system with extra memory capacity.

If, for some reason, you can't (or don't want to) buy a 64, Commodore's VIC-1111 16K Memory Expander may be the ideal solution. It adds 16K of RAM to the VIC-20, increasing the total available RAM to 19.5K.

Using the VIC-1111 16K Memory Expander

The VIC-1111 Memory Expander plugs directly into the VIC-20's expansion port. This port is located on the far left edge of the computer's rear panel as you face the back of the computer.

Follow these steps to insert the cartridge:

1. Turn off the computer and all peripherals.
2. Orient the VIC-1111 cartridge so that its nameplate is facing up.
3. Insert the cartridge into the expansion port of the VIC-20.
4. Press the cartridge firmly into place.
5. Following the proper starting procedures (i.e., computer last), turn on the system.

Be sure not to leave fingerprints on the contacts of the cartridge, since this can interfere with the flow of signals between the computer and the cartridge.

Program Applications for the VIC-1111

The VIC-20 stores each piece of information in a unique memory location. To make the information accessible, each location is labeled with a memory address. This lets the computer know exactly where to look for a specific piece of data. For example, a VIC without memory expansion stores Commodore BASIC programs in memory locations with addresses 4096–7679, filling them from the bottom up. This area represents the 3.5K RAM allotment discussed previously. Because the locations which contain a Commodore BASIC program must be continuous (that is, memory addresses within this area cannot be used for other computing functions), any expansion must be tacked on to either the beginning or end of this area.

Memory locations 1024–4095 are reserved especially for expansion by a 3K RAM memory-expander cartridge, which produces a continuous Commodore BASIC program area from 1024 to 7679. The VIC-20 with no memory expansion, or with only 3K of expansion, uses memory addresses 7680–8191 as screen memory (where screen code values can be POKEd to put characters on the screen without PRINTing them). This placement of screen memory effectively eliminates further allocation of continuous Commodore BASIC program space.

However, the VIC-20 is capable of much greater memory expansion. Inserting the VIC-1111 16K Memory Expander causes the VIC-20 to do extensive memory rearranging so that the computer can offer continuous placement of the full 16K of expanded memory, plus the originally available 3.5K for Commodore BASIC programming.

Three memory areas—the Commodore BASIC program area, screen memory, and color memory—are affected by this memory relocation. The Commodore BASIC program area becomes locations 4608–24575 when the 16K memory expander is in. To be aware of that is particularly important to the programmer who plans to POKE characters or colors onto the VIC's screen.

With the 16K expander in place, the unexpanded VIC's screen memory is used by the new Commodore BASIC program area. To remedy that problem, screen memory is moved to locations 4096–4607. Therefore, you must take special care when entering programs which use the screen memory area and POKE screen graphics. You will need to convert some of these programs so that they will use the new screen memory.

The VIC-20's screen is arranged in 23 rows of 22 columns each. To POKE a character onto the screen's upper left corner (row 1, column 1), the unexpanded VIC requires a POKE of the desired character's screen code into screen memory location 7680. Similarly, on the unexpanded VIC, the last character space on the screen, the lower right corner (row 23, column 22), is screen memory location 8185. However, with the 16K expander in place, these locations become 4096 for the upper left corner and 4601 for the lower right corner.

Generally, if a program is being translated from an unexpanded to a 16K expanded version, you can subtract 3584 from the unexpanded address to determine the expanded address. This is helpful in making the appropriate corrections to previously written programs taken from magazines or books; it will also work with your own programs. However, if you plan to run commercial software in cassette or disk form, problems may arise if the program was designed for the unexpanded VIC-20. It may be necessary to remove the 16K expander before using such programs.

While POKEs to screen memory place character shapes on the screen, similar POKEs to color memory give the POKEd characters their colors. The unexpanded VIC's color memory runs from 38400 to 38911. The screen's upper left corner is represented by color memory location 38400 and the lower right corner is 38905. An additional 16K of memory moves color memory to locations 37888–38399. The upper-left and lower-right screen spaces are represented by 37888 and 38393 respectively. Generally, the rule for converting unexpanded to 16K expanded color memory locations is to subtract 512 from the unexpanded value.

Reference Section

Memory Relocation

Memory Affected	Unexpanded Addresses	16K Expanded Addresses
BASIC program area	4096–7679	4608–24575
Screen memory	7680–8191	4096–4607
Color memory	38400–38911	37888–38399

Screen and Color Memory Conversion Formulas

Memory Area	Formula
Screen memory	Unexpanded screen address − 3584 = Expanded screen address
Color memory	Unexpanded color address − 512 = Expanded color address

Glossary

Nonvolatile memory. Computer memory which is not erased when the computer's power is turned off (ROM).

RAM (random access memory). The type of computer memory that holds the user's programs.

ROM (read only memory). The type of computer memory that holds the computer's built-in instructions. This memory cannot be written to by the user.

Volatile memory. Computer memory which is erased when the computer's power is turned off (RAM).

Chapter 8

The VIC-1211A Super Expander for the VIC-20

Chapter 8

The VIC-1211A Super Expander for the VIC-20

Whatever your programming interests, you have probably discovered the importance of graphics. Obviously, graphics usually make games better. They help educational programs as well, since most students are more likely to utilize educational programs that include interesting visual displays. Even business applications benefit from graphics routines; for instance, financial presentations are much more understandable when charts and graphs supplement the standard spreadsheet and ledger.

The home programmer often wants to try sophisticated graphics—and there is one critical element which makes high-resolution computer graphics possible. That technique is called *bitmapping*. A pixel (a single dot on your TV or monitor) is the smallest controllable graphics unit; in order to directly manipulate each of a computer's pixels, you need to use bitmapping.

In bitmapping mode, each dot on the computer's screen is controlled by its own bit in the computer's memory. Thus, you can alter the screen simply by changing the contents of the memory representing this screen image. The VIC-1211A Super Expander assists you in making the most of that capability.

What Can the VIC-1211A Super Expander Do?

Using high-resolution graphics on the VIC-20 requires extra memory and is made easier by special commands; the VIC-1211A provides both. Not only does this cartridge add an additional 3K of memory (enough to permit bitmapping of the entire VIC screen), but it adds new commands and functions to Commodore BASIC as well. These commands, which control high-resolution plotting, drawing, and coloring, make implementing custom screen graphics as simple as adding a single program line. The cartridge also makes using music and other special-effects programming easier because of the Super Expander's enhanced sound control.

To speed entry of programs, the Super Expander cartridge assigns a different command to each of the VIC's function keys. A single press of a function key prints the entire command word onto the screen.

Using the VIC-1211A Super Expander

To insert the cartridge into the computer, follow these steps:

1. Be sure that the VIC-20 is turned off.
2. Orient the Super Expander cartridge so that its nameplate is facing upward.
3. Insert the cartridge into the VIC-20's expansion port (located on the far left edge of the VIC-20's rear panel as you face the back of the computer).
4. Press the cartridge firmly into place.
5. Turn the system on, making sure that the VIC-20 is turned on last.

The screen will appear as usual, except that it will read 6519 BYTES FREE. This reflects the additional memory that the Super Expander provides.

The one function key which is not assigned a Super Expander command gives you the ability to LIST a program with a single key press. Functions f1, f3, f5, and f7 are accessed by pressing the designated key. Functions f2, f4, f6, and f8 are accessed by holding the SHIFT key while pressing the function key. The assignment of the function keys, as defined by the Super Expander, is given in Table 8-1.

Table 8-1. Super Expander Function-Key Assignments

Function Key	Assignment	Function Key	Assignment
f1	GRAPHIC	f2	COLOR
f3	DRAW	f4	SOUND
f5	CIRCLE	f6	POINT
f7	PAINT	f8	LIST [RETURN]

Conveniently, one of the commands the Super Expander offers is KEY. Entering KEY as a direct command lists each of the function keys' assignments on the screen. Otherwise, KEY can be used to assign a different function to a specified key, ranging from a different command to a string of commonly used text. For example, KEY 3, "SCNCLR" (and then pressing RETURN) would assign the Super Expander command SCNCLR (for screen clear) to function key f3, replacing the key's previous DRAW command.

Controlling CHR$ codes can also be included in the function key assignment, allowing you to assign the KEY command plus a carriage return CHR$ code to one of the function keys. This would allow you to press that particular function key to see a listing of all function key assignments. For example, KEY 7, "KEY" + CHR$(13) (and pressing RETURN) gives the new assignment to function key f7. Thereafter, when f7 is pressed, the KEY command will be carried out immediately, because CHR$(13) is the code for a carriage return. In other words, f7 creates the command and enters it, too.

Program Applications for the VIC-1211A Super Expander

The Super Expander's commands and functions are incorporated into Commodore BASIC, following normal rules of syntax with only one exception. When a Super Expander command is to be the result of an IF/THEN statement, a colon must separate the statement THEN from the Super Expander command. For example, this is the correct syntax for an IF/THEN statement incorporating such a command:

IF condition THEN: Super Expander command

The Super Expander adds ten commands and seven functions to Commodore BASIC. Four of these set the parameters of a graphics program:

Command	Parameter
COLOR	Sets screen, border, character, and auxiliary colors
GRAPHIC	Sets one of five graphics modes
REGION	Sets a new character color
SCNCLR	Clears the entire graphics screen

The first two commands, COLOR and GRAPHIC, must be included in the opening lines of any program that utilizes Super Expander graphics. COLOR is followed by four numbers, separated by commas, which specify screen color (0–15), border color (0–7), character color (0–7 for normal mode or 8–15 for multicolor mode), and auxiliary color (0–15), in that order.

GRAPHIC chooses the type of graphics the program will use and is followed by a single number. The five possible graphics modes are 0 (normal text mode), 1 (multicolor mode), 2 (high-resolution mode), 3 (a combination of high-resolution and multicolor mode), and 4 (which resumes normal text mode).

Normal text mode, as its name implies, allows text to be PRINTed normally on the VIC's screen. This mode's value, 0, is the default value, so text is PRINTed as usual when the VIC Super Expander is first turned on. Mode numbers 1, 2, and 3 cause the entire VIC screen to be automatically bitmapped so that special graphics functions can be used.

Mode 2, the high-resolution mode, allows straightforward high-resolution drawing and plotting on the graphics screen in either the current screen or character color. Mode 1, the multicolor mode, permits the same commands as for mode 2 but offers only half the resolution. However, multicolor mode allows drawing and plotting in any of the four colors selected by the COLOR command.

Mode 3 is a mixture of modes 1 and 2. It can produce either graphics effect, depending on the value of the current character color. A character color from 0 to 7 selects high-resolution graphics, while a value from 8 to 15 sets multicolor graphics.

Finally, mode 4 returns to normal text output from graphics modes 1, 2, or 3. Never select mode 4 while the computer is already set to text mode; this will cause a total system crash which can be recovered only by turning the VIC-20 off and then on again.

The REGION command selects a new character color without changing the screen, border, and auxiliary colors. It takes the form REGION, followed by a character color number (0–15).

To erase the entire graphics screen the SCNCLR command is used. No value accompanies this command.

Once the graphics screen's parameters are set, you can start creating high-resolution images. Use the commands CHAR, CIRCLE, DRAW, PAINT, and POINT. CHAR prints text on the graphics screen. CIRCLE draws a circle or arc with given dimensions. DRAW plots a line between specified points. PAINT colors an enclosed area with the chosen color. And POINT gives points with the given coordinates a specifed color.

Each of these commands must be followed by additional information. Table 8-2 lists the required information and the format in which it must be supplied.

Table 8-2. Required Information for Super Expander Commands

Command	Supplemental data
CHAR	Screen row, screen column, "text"
CIRCLE	Color register number, x center coordinate, y center coordinate, horizontal diameter, vertical diameter, beginning angle of arc, ending angle of arc
DRAW	Color register number, first point x coordinate, first point y coordinate TO second point x coordinate, second point y coordinate
PAINT	Color register number, x coordinate within area, y coordinate within area
POINT	Color register number, x coordinate, y coordinate

These commands do not receive their color information in the usual 0–15 code format. Instead, they take on the current color of the color register you specify (0–4). When in high-resolution mode, color register 0 specifies the screen color, and color registers 1–3 specify the character color. (Remember, only two plotting colors are available in high-resolution mode). However, when multicolor mode is in use, color-register numbers are assigned as follows: 0 = screen color, 1 = border color, 2 = character color, 3 = auxiliary color.

Special considerations. All the x,y coordinates are based on a special numbering system in which the screen is divided into a grid measuring 1024 × 1024. A point with coordinates 0,0 would be located at the upper left corner of the screen, while a point with coordinates 1023,1023 would be found in the screen's lower right corner. When distance values (such as those for the horizontal and vertical diameters of the CIRCLE command) are required, the values are also based on this numbering system.

First, the CIRCLE command requires the color it is to draw in. Next, it needs the x and y coordinates of the circle's center and both a horizontal and a vertical diameter of the circle. Since two diameter values are required, you can create ovals simply by varying one of the diameters. But remember one thing: Because screen plotting with the Super Expander is based on a grid measuring 1024 × 1024 units, and since the VIC's screen is really composed of an unequal grid measuring 22 columns × 23 rows, calculation of true circles will be thrown off. As a result, equal values for horizontal and vertical circle diameters will create an oval rather than a circle.

Values for a beginning and ending angle of an arc are included only when an arc is to be drawn. The numbers used for the arc data are based on a system where the circle is divided into 100 parts, called *gradians*. The 0 starting point is located at the circle's three o'clock position, with 25 located at the six o'clock position, 50 at nine o'clock, and so on.

PAINT is used to fill in an enclosed area with a chosen color. This area may have been created by the CIRCLE command or even by connected lines placed on the screen with the DRAW command. To use PAINT you must specify the color you want to use, as well as the x and y coordinates of any point within the enclosed area. Be sure that the area to be PAINTed is really closed, or your PAINT will leak out over the rest of the screen.

Sound

Along with its many graphics features, the Super Expander also provides enhanced sound capabilities through the use of the SOUND command. SOUND simultaneously controls the VIC-20's four voices (including volume). This command's syntax is as follows:

SOUND note value one, note value two, note value three, note value four, volume setting

A value must be specified for each voice, even if the value is 0, which represents off. Note that tone values range from 128 to 255, while volume values range from 0 (off) up to 15 (maximum). (maximum).

You can achieve more intricate sound effects through programming with the music mode. This mode is entered in a program line with the use of the PRINT statement and the CONTROL (CTRL) and left arrow keys. The PRINT statement is written as usual, and an opening quotation mark is printed. Then, hold down the CRTL key while pressing the left arrow key; this leaves a reverse letter F on the screen. While still within quotation marks, you can use special music mode characters to create music or sound effects. The appropriate commands are identified in Table 8-3.

Most of these single-letter or character commands are simply written on the program line, separated from other commands by spaces. However, each command that requires a number range must have a numeric value immediately following it. To end a string of music information, use a closing quotation mark. Pressing the RETURN key exits the music mode at the end of the program line.

Music commands can be continued on a following line, again within a PRINT statement, and without the need to use the CTRL and left arrow keys again. This is done by ending the previous music line with a semicolon after the final quotation mark.

Table 8-3. SOUND Commands

Command	Action
A	Play note A of the musical scale
B	Play note B
C	Play note C
D	Play note D
E	Play note E
F	Play note F
G	Play note G
O	Choose an octave (1–3, lowest to highest)
P	Print music mode actions on screen
Q	Quit printing music mode actions on screen
R	Rest 1 beat
S	Select a voice number (1–4)
T	Set tempo (0–9, fastest to slowest)
V	Set volume (0–9, off to highest)
#	Precedes a note to make it sharp
$	Precedes a note to make it flat

Function Statements

The Super Expander provides function statements to help you more easily use the new commands it allows. Table 8-4 lists these functions along with the actions they perform. Note that when a function is used as a direct command, it must be preceded by the PRINT statement so that the result will be shown on the screen.

Table 8-4. Super Expander Function Statements

Function	Action
RCOLR(X)	Returns the value of the given color register
RDOT(X,Y)	Returns the color of the given point, with coordinates x,y
RGR(X)	Returns the current graphics mode when given any number from 0 to 255
RJOY(X)	Returns the position of an attached joystick when X is any number from 0 to 255
RPEN(0)	Returns the horizontal screen coordinate of an attached light pen
RPOT(0)	Returns the value of game paddle X
RPOT(1)	Returns the value of game paddle Y
RSND(X)	Returns the value of the given sound register; X is 1–4 for voices and 5 for volume

Demonstrations

Using the Super Expander cartridge is the best way to learn what it can do. The short program below uses multicolor functions to produce the image of a three-dimensional sphere on the computer's screen.

```
1 GRAPHIC 3
2 SCNCLR
3 COLOR 1,1,10,6
4 CIRCLE 3,511,512,345,505
5 CIRCLE 2,527,512,347,490
6 CIRCLE 3,700,512,420,600,25,75
7 CIRCLE 2,700,512,429,575,25,75
8 CIRCLE 3,304,512,423,575,75,25
9 CIRCLE 2,323,512,420,590,75,25
10 CIRCLE 3,770,512,380,745,25,
11 CIRCLE 2,780,512,378,700,25,75
12 CIRCLE 3,240,512,380,710,75,25
13 CIRCLE 2,253,512,380,750,75,25
14 CIRCLE 2,512,505,325,130,0,25
15 CIRCLE 3,512,510,340,130,25,50
16 CIRCLE 3,515,515,350,130,0,25
17 CIRCLE 2,515,505,320,130,25,50
```

Line 1 begins by selecting graphics mode 3, a combination of high-resolution and multicolor mode. Line 2 clears the graphics screen. Colors for the sphere and background are selected in line 3; the screen and border are white, while the character and auxiliary colors are red and blue respectively. Because the number for the character color (10) is greater than 7, multicolor graphics is selected.

Line 4 draws a circle in the color specified by color register 3 (the auxiliary color, blue). The circle's center is located at the center of the screen (coordinates 511,512). The horizontal and vertical diameters of the circle are not equal. They have been adjusted to compensate for the plotting system's distortion of screen coordinates.

Line 5 draws a red circle, shifted slightly to the right of the blue circle. Program lines 6–13 plot vertically oriented orange-segment-like divisions to give dimensionality to the graphics sphere. Because these lines produce arcs, two extra pieces of data, not found in the circle-producing program lines (4 and 5), mark the beginning and ending positions of the arcs. For example, the arcs of lines 6 and 7 are drawn from the bottom to the top of the sphere, as specified by arc parameters of 25 and 75. Lines 14–17 perform the function of providing a sort of equator for the sphere. Each of these arcs is drawn in a segment half the size drawn by lines 6–13 (0–25 or 25–75).

The resulting image appears to have great depth as viewed on the screen, and there is a way to give it even more. Because two complete spheres (one blue and the other red) are drawn on the screen, they can be viewed through ordinary blue/red 3-D glasses to produce a real 3-D effect. If this result is not immediately achieved, a slight adjustment of some of the circle's diameter values should enhance viewing.

Reference Section

Syntax of Selected Super Expander Functions

Action	Command line
Print text on the graphics screen	CHAR screen row, screen column, "text"
Draw a circle or arc	CIRCLE color, x center, y center, x diameter, y diameter, beginning degree of arc, ending degree of arc
Choose graphics screen's colors	COLOR screen color, border color, character color, auxiliary color
Draw a line	DRAW color, x1,y1 TO x2,y2
Select a graphics mode	GRAPHIC mode number
Fill an area with color	PAINT color, x within area, y within area
Turn on a screen point	POINT color, x,y
Pick a new character color	REGION color
Clear the graphics screen	SCNCLR
Play a note or notes	SOUND note value for speaker 1, note value for speaker 2, note value for speaker 3, note value for speaker 4, volume setting

Glossary

Bitmapping. The process of assigning each of the computer's graphics picture elements to its own bit in the computer's memory.

High-resolution graphics. The type of graphics by which each of the computer screen's pixels can be instructed independently.

Low-resolution graphics. The type of graphics produced by a computer in which the smallest controllable graphics units are boxes and lines approximately one-fourth the normal character size.

Pixel. A single computer graphics picture element; a pixel can be on or off (light or dark) and is the building block for all characters on the computer screen.

Resolution. The number of pixels supported by the computer graphics screen.

Chapter 9

The 1600 VICMODEM and the 1650 AUTOMODEM

Chapter 9

The 1600 VICMODEM and the 1650 AUTOMODEM

When two computers communicate over phone lines, whether across town or across country, a modem serves as the connection between computer and telephone. The modem not only connects the computer and telephone together, it translates signals from the computer so that they can be sent over the phone lines. This process is known as *telecommunications.*

As a specialized branch of computer communication, telecommunications offers numerous resources to the computer user, and with the VICMODEM (and in some cases a fee) you can tap those resources with your Commodore computer.

What Can the 1600 VICMODEM and the 1650 AUTOMODEM Do?

The VICMODEM can make your VIC-20 or Commodore 64 seem like a much larger computer. Enormous amounts of information are available through the telephone lines, and the VICMODEM lets you reach it.

There is virtually no restriction on the range of the VICMODEM. The VIC and 64 can communicate with another computer in the same town or in a distant state or country. You can exchange programs, carry on conversations with other computer users, access data bases, or even go shopping with the help of the VICMODEM.

One feature that is rarely stressed is that the VICMODEM allows your Commodore computer to communicate with *any* other computer. You can even go on-line with much larger business computers—for instance, those at your work place. University students can access the mainframe computer on campus while sitting at home, instead of waiting in long lines for a terminal.

All the communications features of the VICMODEM are supported in the 1650 AUTOMODEM, and you can use the

AUTOMODEM with either the VIC-20 or the Commodore 64. The AUTOMODEM works much like the VICMODEM, but is more convenient to use. It has the ability to dial automatically and when used with the Commodore 64 to answer automatically as well.

A Tour of the 1600 VICMODEM and 1650 AUTOMODEM

The most noticeable physical characteristic of the 1600 is the user-port interface slot, which connects the modem to the computer. Like the AUTOMODEM, the VICMODEM will connect to either the VIC or the 64.

The red signal indicator on the side of the modem glows when the modem is receiving data of any kind (even when there is a busy signal from the phone on the other end). The answer/originate switch toggles between the *answer mode,* which is used if your computer is sending signals out as a bulletin board (and is therefore receiving calls from other computers) and the *originate mode,* which is used if your computer is acting as a remote terminal.

Finally, along the VICMODEM's rear panel (opposite the user-port interface slot) is the the modular telephone socket. This is where the telephone line and interface meet. All you have to do is plug in the modular plug from a telephone handset.

The 1650 AUTOMODEM is also designed to use in the user port. However, because of its enhanced capabilities, it has a few more external features than the VICMODEM. The AUTOMODEM has a user-port slot and a red indicator light much like the VICMODEM's.

In addition to an answer/originate switch (labeled with the letters A and O), the AUTOMODEM has several other switches. The data/telephone selector switch, labeled with the letters D and T, is next to the answer/originate switch. It permits you to select modem operation and normal telephone operation. This allows you to leave the modem hooked up all the time, switching between modem and voice mode more easily. To the left of this switch is a jack, labeled PHONE, which connects the telephone to the AUTOMODEM. The modem is then connected to the wall phone jack via the LINE jack located on the far left edge of this side panel.

You'll also notice the half/full duplex switch, labeled with the letters H and F. This switch determines how your computer communicates with another computer; the characteristics of the system that you will be communicating with determine which setting you should use.

Using the 1600 VICMODEM and 1650 AUTOMODEM

Besides the VICMODEM, a telephone line, and your computer, all you'll need to begin telecommunications is the 1530 Datassette. Included with the VICMODEM are two special terminal software packages called VICTERM I (for the VIC-20) and 64 TERM (for the Commodore 64). Both of these programs are on cassette and therefore require a Datassette for loading.

Before you can use the VICMODEM, you must set the answer/originate switch. If you will be placing a call to another computer, network, or bulletin board service, move this switch to the O position (for originate). But when your computer will be receiving a call, set the switch in the A position (for answer).

The AUTOMODEM is also supplied with terminal software on cassette, which makes use of the modem's automatic dialing and answering features. These programs are also called VICTERM I (for the VIC-20) and TERM 64 (for the Commodore 64). VICTERM I is not capable of performing automatic answering functions.

The VICMODEM is probably the easiest Commodore peripheral to use. Here's how to get started:

1. Turn off the computer. If you are using the VICTERM program, remove all cartridges from the computer's expansion port.
2. Set the answer/originate switch to its proper position (usually O).
3. Insert the VICMODEM into the computer's user port.
4. Plug the Datassette into the cassette port.
5. Turn on the computer.
6. Place the terminal software into the Datassette (use VICTERM I with the VIC-20 and 64 TERM with the Commodore 64).
7. Type LOAD and press the RETURN key.
8. After the program has been loaded, type RUN and press the RETURN key.
9. Make the proper menu selections.
10. Dial the desired telephone number.

11. When a high-pitched tone is heard through the telephone receiver, disconnect the handset (DO NOT place it back on the cradle).
12. Connect the coiled cable's plug to the VICMODEM's modular telephone socket. The red indicator light will glow when the connection is complete.

Most of the connection steps are common to both the VICMODEM and the AUTOMODEM. However, connection of the AUTOMODEM requires additional instructions.

1. Turn off the computer.
2. Remove all cartridges from the computer's expansion port.
3. Insert the AUTOMODEM into the computer's user port.
4. Plug the Datassette into the cassette port.
5. Set the answer/originate switch to its proper position (usually O).
6. Select half or full duplex with the H-F switch, depending on the telecomputing service you will be using.
7. Select the D position of the data/telephone switch for use of the modem.
8. Remove the modular plug from the rear of the phone's base unit and place it into the AUTOMODEM's LINE jack.
9. Connect the now-empty modular jack on the rear of the phone's base unit to the PHONE jack of the AUTOMODEM via another modular phone cable.
10. Turn on the computer.

Follow steps 6–8 listed above for the VICMODEM to load the terminal software. But before running the program, decide whether you'll dial the number when prompted by the program or whether the number should first be stored for automatic dialing. Follow these steps for dialing the number when prompted:

1. Run the VICTERM I or TERM 64 program.
2. Wait for the TERMINAL READY prompt to appear, then press the f6 key (SHIFT plus f5).
3. As prompted, make sure that the data/telephone switch is set to D and that the answer/originate switch is set to O.
4. Type in the number after the prompt, omitting the slash (/) and the hyphen (-). Press the RETURN key.
5. Wait for the AUTOMODEM's red light to come on and a TERMINAL READY prompt to appear; these confirm connection to the telecomputing service.

6. Press the computer's RUN/STOP key if the modem light comes on but the prompt is not displayed.

If you make a mistake while typing in a phone number, press the up arrow key to clear the entry. Then you can type in the correct number.

Both the VICTERM I and the TERM 64 programs allow phone numbers to be stored for automatic dialing, but the numbers must be preset before the program is run. To preset phone numbers, first load the terminal software into the computer. Do not run the program yet. LIST the program and a phone-number storage screen will be displayed. The VIC can hold a maximum of five numbers with as many as 30 characters each. Ten numbers of up to 30 characters each can be stored in the Commodore 64.

Move the cursor to the asterisk (*) in the first PRINT statement to enter the number. Enter only the number; do not enter such characters such as the hyphen (-) or number sign (#). If desired, you can include a P character to signify that a pause is required to wait for a dial tone (for instance, if you must wait for an outside line after dialing 9). Press the RETURN key when you have finished entering the number. Follow this process for entering the remaining numbers. Finally, you should save the program with phone numbers onto a separate tape for later use.

When using the preset phone numbers within the terminal program, each of the steps for dialing is performed until the phone-number input prompt is encountered. Rather than typing in the number, type the commercial "at" symbol (@), followed by the designated key for the stored phone number (0 for the first number, 1 for the second, and so on). Press the RETURN key, then follow the same instructions as for manually entering phone numbers.

Automatic answering (which can only be done with the Commodore 64) is mainly a process of preparing the modem to wait for a call. Be sure that the AUTOMODEM is installed and that the terminal program is loaded and running. Press f8 (SHIFT plus f7). Follow the prompts, setting the data/telephone switch to D and the answer/originate switch to A. The Commodore 64 will then wait until it receives a call. When a call has been received, the TERMINAL READY screen will appear and the AUTOMODEM's light will come on.

The answering process can also be performed manually by loading and running the terminal software while setting the data/telephone switch to T and the answer/originate switch to A. When you hear the telephone ring, set the data/telephone switch to D and watch for the red light to come on.

Software

Two menus are provided by VICTERM I and 64 TERM; the first is for format and the second is for control.

The format menu (menu 1) lets you determine how the VICMODEM will transmit and receive information for the computer. You can preset baud rate, duplex, parity, stop bit, and word length. Each of these parameters, discussed below, has a distinct effect on the data flow.

Baud rate is the speed of data transmission and reception. The VICMODEM and AUTOMODEM can be set between 0 and 300 bps (bits per second). Even though the menu displays 1200- and 2400-baud options, they should not be chosen.

Duplex is an error-checking system in which letters are mirrored back to the computer. Either full or half duplex can be selected.

Parity is another error-checking system in which selected bits are verified in transmission. None, even, odd, mark, or space parity can be used.

A *stop bit* is a blank bit inserted between characters. The VICMODEM allows one or two stop bits.

Word length is the number of bits that make up a character. Five-, six-, seven-, or eight-bit characters can be selected.

After setting the format, select the control menu (menu 2), which allows you to set the configuration of the computer. The choices on this menu include linefeed, ASCII conversion, two-color option, and format end-of-line.

Linefeed signals the end of a line. Either a simple carriage return or a linefeed (a linefeed plus a carriage return) can be selected.

What kind of computer will your Commodore computer be conversing with? If it's another Commodore computer, select VIC-to-VIC (or 64-to-64). If it is any other type of computer, *ASCII conversion will be necessary.* Use VIC-to-ASCII or 64-to-ASCII depending on your computer.

Two-color option lets your screen show characters sent in a different color from those received. This is either on (selected) or off (not selected).

The *format end-of-line* option allows you to use or eliminate word wrap. This feature prints words, which would normally be split by the screen's margin, on the next line without breaking them. It is either enabled (on) or disabled (off).

When you have finished the selection process, press T to return to an active status (on-line). Even if your Commodore computer is engaged in an active conversation with another computer, the menu-selection process can be called by pressing f4 (SHIFTed-f3). The VICMODEM automatically sends a special WAIT character to the main computer so that no data will be lost while looking at the menus. Once the T is pressed, data transmission and reception can resume.

The VIC-20 screen colors can be changed with VICTERM I. To alter the background color, hold down the CTRL key while pressing f3 key. To change the border color, press the pound sign (£) while holding down the CTRL key. Press CTRL and f5 to change the character color. Finally, if the VICMODEM is in the two-color option mode, press f1 and CTRL to change the received character's color.

The colors on the Commodore 64 screen can be changed with 64 TERM. To change the border color, hold down the CTRL key and press the 1 key. To select a different character color, press CTRL and the 5 key. The background color can be changed by pressing the 8 key while holding down the CTRL key. Finally, if the VICMODEM is in the two-color option mode, the received character's color can be altered by pressing the 4 key and CTRL.

Program Applications with the 1600 VICMODEM and 1650 AUTOMODEM

Since terminal software is included with the VICMODEM, there will probably be little need for any programming. But you can control the modem with your own programs, which could adjust all five of the parameters in the VICTERM I (or 64 TERM) format menu (baud rate, duplex, parity, stop bit, and word length).

These five functions are set with the use of two computer memory registers. The baud rate, stop bit, and word length are set with what is called the control register; parity and duplex are controlled by another memory location, the command register. Each of these registers consists of a single byte (eight bits). Within the byte, selected bits are assigned to the five functions. By determining the condition of each function, a bit pattern for the two register bytes is formed. This final pattern is then translated into a decimal number for use in programming.

In the control register, the decimal values for baud rate, stop bit, and word length are given a fixed number. Table 9-1 shows these values. The number placed in the control register is the sum of the desired function's register values. For example, a 300 baud rate, one stop bit, and a five-bit word length would give a final control register number of 102.

Likewise, parity and duplex are determined by the command register.

The corresponding register values are shown in Table 9-2.

Table 9-1. Control Register Values

Control Register	
Baud Rate	**Register Value**
50	1
75	2
110	3
134.5	4
150	5
300	6
Stop Bit	**Register Value**
1 stop bit	0
2 stop bits	128
Word Length	**Register Value**
5 bit	96
6 bit	64
7 bit	32
8 bit	0

The command register value is also the sum of all the desired function's register values. However, in practice, the command register can be eliminated; its use is optional. This means you would not use parity and would operate the VICMODEM at full duplex. A quick look at the register values for these choices yields a total value of zero. Therefore, the final value of zero would have to be placed into the command register and is obviously irrelevant.

Table 9-2. Command Register Values

	Command Register
Handshake	**Register Value**
XOFF	0
XON	1
Parity	**Register Value**
None	0
Odd	32
Even	96
Mark	160
Space	224
Duplex	**Register Value**
Full	0
Half	16

The VICMODEM uses these two registers when creating a special communications channel, which is used for opening a link between the Commodore computer and the modem. A generalized syntax for creating this channel would be:

OPEN file number, device number, control register value, command register value

The file number for the link ranges from 1 to 255, with `7 127 to 255 performing automatic linefeeds after a carriage return. The device number for the VICMODEM is always 2.

Here is an example that makes a communication link (OPEN 7) with the VICMODEM (2), sets the baud rate to 300, uses two stop bits, has an eight-bit word length (CHR$(134)), uses space parity, and sets full duplex (CHR$(224)):

OPEN 1,2,CHR$(134)CHR$(224)

In this example, the control and command registers are set with character codes. Each character code represents one of the registers. Again, if parity is not needed and full duplex is acceptable, the command register character code could have been removed. Then the line would look like this:

OPEN 1,2,CHR$(134)

With the communications link established, data can be either transmitted or received. Two general commands are used in talking and listening. PRINT# is for transmitting and GET#

is for receiving. A generalized syntax for each of these commands is:

PRINT# file number, data
GET# file number, string variable

PRINT# acts just like the PRINT statement, except that in this case all of its data is sent via the modem to the computer on the other end of the telephone line.

Finally, on the conclusion of all data transmission and reception, you must sever the communications link. The CLOSE command disconnects the previously established file. A generalized syntax follows:

CLOSE file number

In order to work properly, the file number must correspond with a file that has already been OPENed. Using the CLOSE command is not necessary, however, if the Commodore computer will be turned off immediately after the VICMODEM has been used; in that case the file will be CLOSEd automatically.

Reference Section

Syntax of VICMODEM Functions

A generalized syntax is given for each function.

Function	Command
Close a file	CLOSE file number
Open a file	OPEN file number, device number, control register, command register
Receive data	GET# file number, string variable*
Transmit data	PRINT# file number, data*

* An OPEN command is required prior to these functions and a CLOSE afterward. *Do not include the* asterisk (*) *as part of the command.*

Special VICTERM I and 64 TERM Menu Functions

Menu 1

Function	Command
Set baud rate	Press B, then right cursor to set correct rate and press RETURN
Set duplex	Press D, then right cursor to set full or half duplex and press RETURN
Set parity	Press P, then right cursor to set parity type and press RETURN
Set stop bit	Press S, then right cursor to set bit number and press RETURN
Set word length	Press W, then right cursor to set word size and press RETURN

Menu 2

Function	Command
Set linefeed	Press L for linefeed and carriage return or press C for just carriage return
Set VIC-to-VIC	Press V to talk with another Commodore computer or press A to talk with any other type of computer
Set two-color option	Press 2 to toggle two-color option on and off
Set format end-of-line	Press F to toggle format end-of-line on and off

Both Menus

Function	Command
Move between menus	Press N
Move to menus from active status	Press f4 (SHIFTed-f3)
Return to active status	Press T

Glossary

ASCII. *A*merican *S*tandard *C*ode for *I*nformation *I*nterchange; a standard for coding characters.

Baud rate. Speed of data transmission measured in bits per second (bps).

BBS. An electronic bulletin board service for public access via modem.

Data base. In telecommunications, a private or subscription-service information library; also a program that creates such a collection of data.

Duplex. An error-checking feature for data transmission in which the data is repeated back to the sender. Can be either full or half duplex.

Host. The computer receiving calls (requests for information) in a BBS or other information service.

Modem. *Mo*dulator-*dem*odulator; a peripheral device for connecting a computer to a telephone line.

On-line. An active state or condition in which the computer and its peripherals are all able to function properly. This term is also used to describe the time spent connected to a host computer (time for which there is usually a fee).

Parity. An error-checking feature for data transmission in which selective bits are manipulated and then examined by the receiver. Can be even, odd, mark, or space.

RS-2323C. An important serial interface standard.

Serial. Data transmitted or received a bit at a time over a single line.

Stop bit. Signals the end of a character. Can be one or two stop bits.

Terminal. Usually, a computer that is remote from the main computer. With a modem, this would be the computer that is making the telephone call to request information.

Word length. The number of bits constituting a given character. Can be five, six, seven, or eight bits.

Chapter 10

CP/M for the Commodore 64

Chapter 10

CP/M for the Commodore 64

One of the most complex tasks that a computer must handle is disk drive operation. Besides understanding all the special drive commands, the computer must also be able to handle data storage and retrieval from the floppy disk. Fortunately, the disk operating system (or DOS) takes care of the disk drive functions, allowing the computer user to sit back and make full use of disk data storage without a great deal of effort.

This approach to control is not restricted to peripherals, such as disk drives, however; other operating systems exist for different jobs. Without a doubt the most popular of these other operating systems is CP/M (control program for microcomputers). CP/M was originally designed as strictly a business-oriented operating system. However, its widespread acceptance by computer manufacturers helped create an immense library of software.

The CP/M system is commonly used by a wide variety of commercial software. While several versions of CP/M exist, the most common one is geared toward the Z80 family of microprocessors (this includes the 8080 and 8085 microchips). Programming with this version closely resembles programming in assembly language; it is this feature that lends almost a language quality to CP/M.

But as with every operating system, information management is its most vital concern. CP/M stores data on floppy disks through an elaborate file-organization process. Each file is stored on disk with a unique name. Then, whenever a task is to be performed on a particular file, that file is referred to by its specific name.

As an operating system, CP/M consists of several useful utilities. Most of these allow data movement, reconfiguration of the computer's memory, and communication with peripheral devices. However, one of the more flexible of these utilities can copy files from a floppy disk into the computer's memory. From there files can be processed to a different peripheral device or copied to another floppy disk.

CP/M does away with much of the input/output drudgery. It is also an efficient manager of the computer's resources and can be modified to suit a variety of computers. But by using this standard operating environment, software authors are able to design programs that are compatible with many different computers. In fact, this is the real virtue of CP/M: it provides a smart operating system, thereby allowing the computer itself to concentrate on computing. So, while CP/M is not necessarily the *best* operating system available, it is nevertheless one *widely accepted* DOS.

CP/M for the 64 is a sophisticated package. Because it would take a whole book to fully cover its features, we'll look at only a few in this chapter, just to give you an idea of how easy it is to use with the 64. Keep in mind that CP/M is an alternative operating system for the 64, offering more advanced features than the standard DOS Commodore provides in its disk drives. (For more details see *COMPUTE!'s Reference Guide to CP/M on the Commodore 64.*)

What Can CP/M for the Commodore 64 Do?

The most obvious asset of CP/M for the 64 is compatability with CP/M-based programming. CP/M requires the use of a Z80 microprocessor, and CP/M for the Commodore 64 has a Z80 microprocessor cartridge. When the cartridge is activated by CP/M, the 64 is ready to perform user programming under the direction of CP/M. Therefore, the Commodore 64 can be made into a dual microprocessor computer by the CP/M and the Z80 cartridge.

Using CP/M for the Commodore 64

Using CP/M on the 64 requires two pieces of equipment: the Z80 microprocessor cartridge and Digital Research's CP/M system disk. Both of these items are used simultaneously when operating CP/M.

Before using CP/M, read and sign the Digital Research License Agreement. This agreement also serves as the registration card. While many people ignore registration and warranty cards, Digital Research provides a newsletter as an added incentive to return this one. That newsletter, plus the other Digital Research product announcements, are worthwhile to any CP/M user, especially the Commodore 64 owner.

Because there are no moving parts or switches on the Z80 microprocessor cartridge, installation of CP/M is remarkably easy. Here is the procedure:

1. Turn off the Commodore 64.
2. Push the Z80 microprocessor cartridge into the 64's cartridge slot (on the left-hand side of the rear panel, as you face the rear).
3. Connect all necessary peripherals (e.g., the 1541 Disk Drive, 1525 Graphic Printer, etc.).
4. Turn on all other peripherals.
5. Turn on the 64.
6. Place the Digital Research CP/M system disk in the 1541 Disk Drive.

CP/M for the Commodore 64 is designed to operate with all of Commodore's peripherals. A special CONFIG (configuring) utility permits user selection of a variety of Commodore products. However, other manufacturers' printers and disk drives may also be used, although custom programming will be necessary for patching with CP/M.

Once the CP/M system disk is placed in the disk drive, it must be loaded and run. Here is the procedure:

1. With the initial BASIC message, BYTES FREE, present, type LOAD "CPM",8 and press the RETURN key.
2. When the READY message appears, type RUN and press the RETURN key.
3. After 27 asterisks (*) have been printed across the screen, the CP/M message will appear.
4. CP/M is ready when the CP/M prompt (A>) appears.

At that point, you can access CP/M files and utilities, run CP/M applications, or write CP/M programs. There is one thing to remember, however: You must use disks formatted for use with the Commodore computer. This does not create any problems if custom programs are being designed, since the disks will have to be formatted with the Commodore system anyway. But if you want to use prepackaged CP/M software, they must be specially designed for the Commodore format. This is dictated by the nature of the Commodore disk drives. Make sure that a prepackaged CP/M application has been specially prepared in the Commodore format before you purchase it.

Program Applications with CP/M for the Commodore 64

After CP/M has been loaded into the 64, a wide range of activities are possible. The CP/M utilities are among the more versatile features that are available.

Of these utilities, COPY, CONFIG, and MOVCPM may be the most frequently used. These three utilities can format a disk, copy a disk, change printer types, change disk drive types, or create a different-sized CP/M version. In each case, leave the CP/M system disk in the 1541 Disk Drive until you are instructed to remove it. Failure to follow this procedure will crash the CP/M system, with full recovery possible only through a complete restart. But be careful to remove the system disk at the proper time, or it could be erased and CP/M lost forever. You can protect the disk with a write-protect tab, (the tabs are easy to remove whenever you want to write to the CP/M disk). Commodore recommends that you IMMEDIATELY make a backup copy of your CP/M disk and that you store the original in a safe place away from damaging magnetic fields.

COPY. When the CP/M prompt (A>) appears, you may access COPY by typing COPY. The system disk must be in the drive when you start this utility.

After it is loaded, the utility will display a menu. Two of the most common menu selections are those which format a disk and back up a disk (COPY a disk).

Here is the procedure for formatting a CP/M disk:

1. Choose option 1 by pressing 1.
2. REMOVE THE CP/M SYSTEM DISK NOW!
3. Place the disk to be formatted in the disk drive and then press the RETURN key.
4. When the formatting is complete, either format another disk or return to the main menu.
5. If no more disks are to be formatted, place the system disk back in the disk drive.

To make a backup copy of a CP/M disk, you need a Commodore CP/M preformatted disk. When using the backup option, the menu will refer to a master disk and a slave disk. These refer to the disk to be copied and the disk to be copied on respectively. Don't confuse the two disks. If you do, there is a chance you will lose some of your data.

Follow this procedure to make a backup disk:

1. Choose option 2 by pressing 2.
2. REMOVE THE CP/M SYSTEM DISK NOW!
3. Insert the MASTER disk.
4. Press the RETURN key.
5. Remove the MASTER disk and insert the SLAVE disk.
6. Press the RETURN key.
7. Repeat steps 3–6 until copying is complete.

When you finish the backup option, the COPY menu will reappear. The CP/M system disk should be placed back in the disk drive at that time.

CONFIG. Although CP/M for the Commodore 64 is initially set for using the 1525 Graphic Printer and the 1541 Disk Drive, other Commodore peripherals can also be added. The CONFIG utility permits the user to choose between two different types of Commodore printers and two different types of Commodore disk drives.

Make sure that the system disk is in the disk drive. When the CP/M prompt (A>) is present, type CONFIG and press RETURN. The I/O configuration menu will appear. Several options are available with this utility, but of major importance is the selection of printer type and disk drive type.

To change the printer type, for example, choose option 2 by pressing 2. CP/M is then configured for Commodore 4022-type printers. To switch back to a 1525-type printer, just choose option 2 again. This selection process toggles back and forth between those two Commodore printer types.

To change the disk drive configuration, choose option 1 by pressing 1. This option toggles back and forth between the Commodore 1541 Single Disk Drive and the Commodore 4040 Dual Disk Drive. You can use either with CP/M on the 64, but Commodore serial devices (for example, the 1541 Disk Drive and the 1525 Graphic Printer) cannot be CONFIGured with Commodore IEEE devices (for instance, the 4040 Dual Disk Drive and the 4022 Printer). As a precaution against any mismatch, recheck the I/O configuration menu before exiting the CONFIG utility.

MOVCPM. The CP/M memory size can be altered to fit many different circumstances. The CP/M system disk comes as a 44K version. However, the most popular memory size among CP/M applications is 48K. The MOVCPM utility moves the CP/M operating system to fit into any memory size specified by the user (with a maximum limit of 48K).

To create a 48K CP/M version, type MOVCPM 48 * and press RETURN. This executes the relocation and places the different CP/M version in the computer's memory. Two other utilities, SAVE and SYSGEN, can now act on the new version of CP/M. SYSGEN is the more commonly used utility, because it writes the new version on the system tracks of a floppy disk.

To use SYSGEN after MOVCPM, type SYSGEN and press RETURN. A prompt will appear on the screen, asking for the source drive's name. This question should be skipped (by pressing the RETURN key). But when the prompt for the destination drive's name appears, type A and press RETURN. Make sure that you have removed the main CP/M system disk from the disk drive.

In order to properly save a copy of this new version of CP/M, you must have a correctly formatted CP/M disk in the disk drive before the destination drive name response. If you want verification of this relocated version of CP/M, you must turn off the 64 and then turn it on again. Then, following the same starting procedures, load the new version instead of the original CP/M system disk.

Reference Section

Syntax of Commodore CP/M Functions

Function	Command
Change disk drive type	CONFIG, press 1
Change printer type	CONFIG, press 2
Copy a disk	COPY, press 2, insert MASTER disk, replace with SLAVE disk, repeat until finished
Create a new CP/M version	MOVCPM new memory size
Format a disk	COPY, press 1

Glossary

BDOS. *B*asic *D*isk *O*perating *S*ystem interface for a program to the operating system.

BIOS. *B*asic *I*nput *O*utput *S*ystem; supplies the BDOS with input/output operations.

CP/M. Control *P*rogram for *M*icrocomputers operating system for providing a standard interface for Z80-microprocessor-based computers. Also aids the computer in controlling its peripherals.

I/O. *I*nput/*O*utput of data through the computer's peripherals.

MP/M. *M*ulti-*P*rogramming *M*onitor control program for dealing with multiple terminals, each performing multiprogramming.

Operating System. A group of programs for controlling a computer's peripherals and running its programs.

Z80. An eight-bit microprocessor used in conjunction with CP/M.

Index

A Complete System

Soon after purchasing your VIC-20 or Commodore 64, you undoubtedly discovered that home computers are sophisticated and powerful tools. But you've also found that, by itself, the computer is not enough.

With nothing but a computer, how can you use software on tape or disk? How can you print out a copy of a program listing or even a letter? How can you access data bases, save programs to tape, or use non-Commodore devices?

What you need are *peripherals*. But which peripherals? What do they do? How do you hook them up, and how do you use them once they're connected? *Commodore Peripherals: A User's Guide* answers those questions. It's the first comprehensive guide to the myriad of Commodore peripherals available for the VIC and 64. From Datassettes to modems and interfaces, it clearly explains each device and tells you what it can do.

The peripherals documented in this book include:

- 1541 disk drive
- VIC-1525 graphic printer
- VIC Super Expander
- CP/M for the Commodore 64
- Model 1600 VICMODEM
- 1530 Datassette
- And others, from memory expanders to printer/plotters.

Would you like to save a program or run some commercial software? Want to print out that letter? Find out about memory expanders and modems? *Commodore Peripherals: A User's Guide* explains the necessary peripherals in clear, easy-to-understand terms. It will help you turn your computer into an even more powerful tool.

ISBN O-942386-56-6